

# Microsoft Band Web Tile

## Web Tile Documentation

By using this Microsoft Band Web Tile, you agree to be bound by the [Terms of Use](#). Further, if accepting on behalf of a company, then you represent that you are authorized to act on your company's behalf.

<b>1</b>	<b>Overview.....</b>	<b>2</b>
1.1	Introduction .....	2
1.2	Features .....	2
1.2.1	Multi-Platform support .....	2
1.2.2	Tile creation, installation and management.....	2
1.2.2.1	Web tile .....	3
1.2.2.2	Installation .....	4
1.2.2.3	Management .....	4
1.2.3	Web resource push to the Band.....	4
1.2.4	Tile notifications .....	4
<b>2</b>	<b>Web tile details and examples .....</b>	<b>5</b>
2.1	Web tile package architecture .....	5
2.1.1	manifest.json .....	5
2.1.2	Icon files .....	5
2.2	Manifest file contents.....	5
2.2.1	manifestVersion (required) .....	6
2.2.2	name (required) .....	6
2.2.3	description.....	6
2.2.4	version .....	6
2.2.5	versionString.....	7
2.2.6	author .....	7
2.2.7	organization.....	7
2.2.8	contactEmail.....	7
2.2.9	tileIcon (required).....	7
2.2.10	badgelcon.....	7
2.2.11	tileTheme .....	7
2.2.12	refreshIntervalMinutes .....	8
2.2.13	resources (required) .....	8
2.2.14	icons .....	9
2.2.15	pages (required).....	9
2.2.16	notifications .....	10
2.3	Predefined layouts .....	11
<b>3</b>	<b>Writing web tiles.....</b>	<b>13</b>

<b>3.1</b>	<b>Getting started .....</b>	<b>13</b>
<b>3.2</b>	<b>Creating a “feed” web tile.....</b>	<b>15</b>
<b>3.3</b>	<b>Creating a “simple” web tile with multiple resources and pages .....</b>	<b>17</b>

# 1 OVERVIEW

---

## 1.1 INTRODUCTION

Microsoft Band Web Tiles make it extremely simple to deliver relevant information to the Band from virtually any data source accessible through the web. Developers only have to write web tile code once to support multiple mobile platforms (iOS, Android and Windows) since the Microsoft Health app takes responsibility for doing the rest – installing the web tile to the Band, accessing web resources for content, extracting and formatting the content, and delivering that content to the associated web tile on the Band.

A web tile, fully described by a “web tile package”, contains a variety of information including tile metadata, information needed to fetch data from the web and format it for the tile, and other resources associated with creation and updates of the tile.

## 1.2 FEATURES

The features offered by Microsoft Band Web Tile are as follows:

- Multi-platform support
- Tile creation, installation and management
- Web resource push to the Band
- Tile notifications

The following subsections describe these features in more detail.

### 1.2.1 Multi-Platform support

Microsoft Band Web Tile is supported on a wide range of platforms. This includes:

- Windows Phone 8.1 and later
- iOS 7 and later
- Android 4.2 (API 17) and later

### 1.2.2 Tile creation, installation and management

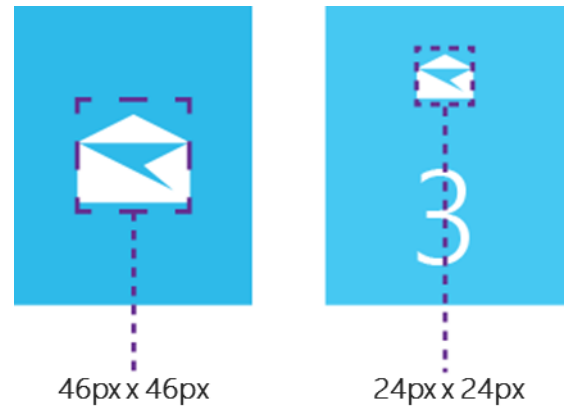
Developers can create a web tile (which looks and behaves just like a tile created using [Microsoft Band SDK](#)) on the Band by building a web tile package and opening it on a mobile device with the Microsoft Health app installed. Once delivered as a file or via HTTP, a web tile package will invoke the Microsoft Health app to start the installation process.

### 1.2.2.1 Web tile

Each web tile is visually represented on the Start Strip by an icon that fits within a 46px x 46px box.

Developers can create two types of web tiles on the Band:

1. Web tiles specialized on handling a feed resource
2. Web tiles for general web resources



A web tile specialized on handling a feed resource can only point to a single web resource that could be either RSS or ATOM data. The tile may contain up to 8 pages of the same layout populated with the latest feeds. If this type of web tile uses the simple layout with a title and a wrapped body, it can show a badge count as new content arrives. When badged, a small 24px x 24px icon (if provided) will be used to accommodate the badging system. If no small icon was provided, the badging functionality will be disabled for that specific tile.



*Three new pages were created, updating the badge count to be 3.*

With web tiles for general web resources, developers control exactly how many pages to show inside of a tile (up to 7 pages), how many web resources to point to and content of individual pages. They can set the content of a page that was created during installation at each update, unlike feed-specialized web tiles where every new feed data item results in the creation of a new page inside the tile. Once a feed page is added to the tile, it is never changed.



*A web tile with two pages with different layouts*

Refer to the [detailed section on web tile creation](#) for more information on how to create a web tile.

### 1.2.2.2 Installation

After a web tile package is created, it can be delivered as a mail attachment or can be accessed as a file saved locally or in one's cloud storage. Opening a web tile package on a device with the Microsoft Health app installed will automatically launch the app and trigger the installation process.

A web tile package can also be accessed via an HTTP hyperlink. For this case, a custom URL scheme must be used (mshealth-webtile) to activate the installation workflow in the Microsoft Health app when the link is referenced. The full custom URL for a web tile will look like this:

```
mshealth-webtile://?action=download-manifest&url=http://www.microsoft.com/mywebtile.webtile
```

### 1.2.2.3 Management

Once a web tile is successfully installed on the Band, it will be listed under *Tiles you added* in the **Manage Tiles** page of the Microsoft Health app. Users can later uninstall web tiles from this page.

### 1.2.3 Web resource push to the Band

Web tile content can be populated periodically with new web resources that are either in JSON or XML format which includes ATOM and RSS feeds. Developers have control over exactly which part of a web resource to show inside a web tile and how often data refresh should happen. The default data refresh interval is 30 minutes and the minimum is 15 minutes but the actual rate is affected by mobile phone connectivity.

Refer to the [detailed section on web tile resource assignment](#) for more information on how to assign web resources to a web tile.

### 1.2.4 Tile notifications

Developers can set up dialog notifications for a web tile which get delivered to the Band when predefined conditions are detected in the web data. Dialog notifications are popups that are meant to quickly display information to the user. Once the user dismisses the dialog, the information contained therein does not persist on the Band. Notification conditions get evaluated against the new, unseen web data for every data refresh.

Web tile supports condition expressions with the following comparison operators for invoking notifications. Unless otherwise specified, the operators support both string (case-sensitive) and numeric operands.

Operator (case sensitive)	Description
==	Returns true if the operands are equal.
!=	Returns true if the operands are not equal.
>=	Returns true if the left operand is greater than or equal to the right operand.
<=	Returns true if the left operand is less than or equal to the right operand.
>	Returns true if the left operand is greater than the right operand.
<	Returns true if the left operand is less than the right operand.
contains	Returns true if the right operand is a substring of the left operand. Requires both operands to be strings.
true	No operands accepted.

When comparing a number and a string, the string is converted to a number value. If the number conversion fails, the '!=' operator returns true and all other operators return false.

Refer to the [detailed section on web tile notification](#) for more information on how to create notifications.

## 2 WEB TILE DETAILS AND EXAMPLES

### 2.1 WEB TILE PACKAGE ARCHITECTURE

A web tile package is a collection of files created using the standard “zip” format. Web tile packages should be renamed with a “.webtile” file extension. The files contained in the web tile package are described in the following sections. References to files within the package are always relative to the root of the zip archive.

Here’s the directory structure of a typical web tile package file:

```
mytile.webtile:
    /manifest.json      // Contains web tile definition and references to other assets
    /icons/*.png        // PNG icons used in the web tile
```

#### 2.1.1 manifest.json

The manifest file must be named “manifest.json” and lives in the root of the archive. The manifest contains information about the web tile and references to the other assets contained in the archive. Refer to the [detailed section on manifest authoring](#) for more information on how to create a manifest file.

#### 2.1.2 Icon files

The web tile package contains at least one icon PNG file, for the tile itself. A smaller version of the icon may also be provided if the web tile has badging enabled. Additional icons may be provided to be used in the tile pages.

A section within the manifest lists all of the icons referenced by the web tile so that they can be pre-loaded onto the Band when the web tile is first installed.

Icon files are typically placed under an “icons” folder and must be PNG files. When pushed to the Band device, PNG images will get flattened to one solid color based on alpha values.

### 2.2 MANIFEST FILE CONTENTS

The manifest.json file contains a variety of information used by the Microsoft Health app to install and manage a web tile. The file is formatted as a single JSON object. The members of the object are listed in the following table:

Member Name	Type	Required	Description
<b>manifestVersion</b>	Number	Yes	Defines the contract between the web tile developer and the Microsoft Health app. It must be set to 1 for the current release.
<b>name</b>	String	Yes	Defines the name of the web tile.
<b>description</b>	String	No	Provides description about the web tile.
<b>version</b>	Number	No	Indicates the version number of the web tile which will not be shown to users.
<b>versionString</b>	String	No	Indicates the web tile version as shown to users. If not provided, it displays the <i>version</i> .
<b>author</b>	String	No	Specifies the author of the web tile.

<b>organization</b>	String	No	Specifies the organization associated with the web tile.
<b>contactEmail</b>	String	No	Provides contact info of the web tile author. Used for administrative purposes and analytics by Microsoft Band team.
<b>tileIcon</b>	String	Yes	Specifies the location of the large tile icon.
<b>badgelIcon</b>	String	No	Specifies the location of the small tile icon. If not provided, the badging functionality is not enabled.
<b>tileTheme</b>	Object	No	Specifies the custom theme colors for the web tile.
<b>refreshIntervalMinutes</b>	Number	No	Specifies the desired data refresh interval, in minutes. Refresh intervals shorter than 15 will not be honored. If not provided, it defaults to 30.
<b>resources</b>	Array of objects	Yes	Each web tile must refer to one or more web resources. Each resource object describes a web data from which content for the tile will be extracted.
<b>icons</b>	Array of strings	No	All PNG icons used by the tile within its pages must be provided here.
<b>pages</b>	Array of objects	Yes	Each tile consists of one or more pages, as described by the objects in this array. Each page object specifies the layout and content of the page.
<b>notifications</b>	Array of objects	No	Creates notification that gets delivered when certain conditions are detected in the new web data.

The members of the object are described, with examples, in the following sections.

#### 2.2.1 manifestVersion (required)

This member defines the contract between the web tile developer and the MS Health app, allowing for future changes to the manifest file contents. For the current release, the value must be the number 1.

```
"manifestVersion" : 1
```

#### 2.2.2 name (required)

The short name of the web tile (max 21 characters). This name is used in the Microsoft Health app.

```
"name" : "my web tile"
```

#### 2.2.3 description

The description is optional field to provide detailed information about the web tile. The description may contain up to 100 characters.

```
"description" : "what's cool about my web tile"
```

#### 2.2.4 version

This is a numeric field that specifies the version number of the web tile. It must contain a positive integer that is incremented whenever a new version of the tile is created. If not provided, it defaults to 1.

```
"version" : 3
```

### 2.2.5 versionString

The versionString member is a friendly form of the version number. If present, it will be used when displaying version information. If versionString is absent, the version number will be shown. The maximum number of characters is 10.

```
"versionString" : "1.3 beta"
```

### 2.2.6 author

Provides the web tile author's name. The maximum number of characters is 50.

```
"author" : "John Smith"
```

### 2.2.7 organization

Provides information about the organization associated with the web tile. The maximum number of characters is 100.

```
"organization" : "Microsoft Band"
```

### 2.2.8 contactEmail

Provides an email address for the web tile author. This may be used for administrative purposes and analytics. The maximum number of characters is 100.

```
"contactEmail" : "john@contoso.com"
```

### 2.2.9 tileIcon (required)

Each web tile must contain an icon that will be used to represent it on the Band Start Strip. This member specifies the location of the large tile icon, using a mapping from icon size to icon location. It must be 46 x 46 pixel PNG icon with transparent background.

```
"tileIcon" : {
  "46" : "icons/largeIcon.png"
}
```

### 2.2.10 badgeIcon

For web tiles that use badge counts, a smaller "badge" icon must be specified. Like the tileIcon, a mapping from size to location is used. It must 24 x 24 pixel PNG icon with transparent background.

```
"badgeIcon" : {
  "24" : "icons/smallIcon.png"
}
```

### 2.2.11 tileTheme

If present, the tileTheme overrides the user-selected theme colors for the web tile's pages. The tileTheme member is an object containing all overrides for individual theme colors. If there are any non-specified theme colors, all the specified values will be ignored and the current band theme will be used instead.

The format of the color values is "RRGGBB" where RR is the hex value for red, GG for green, and BB for blue. Refer to [Microsoft Band Experience Design Guidelines](#) for more information on base, highlight, lowlight, secondary, high contrast and muted.

```
"tileTheme": {
```

```

    "base": "d94c66",
    "highlight": "ea5475",
    "lowlight": "c64763",
    "secondary": "a3919c",
    "highContrast": "bf455f",
    "muted": "993344"
  }

```

### 2.2.12 refreshIntervalMinutes

Specifies the desired refresh interval for the tile, in minutes. If present, this must be a positive integer value. The Microsoft Health app will make a best effort to refresh the tile at the desired rate. Refresh intervals shorter than 15 will not be honored. If not provided, it defaults to 30 minutes. The minimum refresh rate of 15 minutes is not always guaranteed because the Microsoft Health app background sync is subject to OS time quotas.

```
"refreshIntervalMinutes" : 90
```

### 2.2.13 resources (required)

The resources section enumerates the web resources that provide content for the web tile. For each web resource, a resource description object contains:

- **url** – the URL to be referenced. For resources that require authentication, the URL must be HTTPS.
- **style** – specifies the kind of web resource being accessed. Permitted values are “simple” and “feed”. The default value is “simple”.
  - “simple”: Web tiles for general web resources
  - “feed”: Web tiles specialized on handling a feed resource
- **content** – this member specifies how to extract content from the web response. It consists of an object in which each member name defines a “variable” in the web package and the member value is an expression (e.g. xpath) describing where the content can be found in the response. Variable names can use alphanumeric characters and ‘\_’. But the first character of the name cannot be a digit. The expression is interpreted according to the content type of the response. For XML content, the expression is an xpath. For JSON content, the expression is a simplified JavaScript object expression.

*Example with “simple” web resource in JSON format:*

```

"resources": [
  {
    "url": "http://services.faa.gov/airport/status/SEA?format=json",
    "style": "Simple",
    "content": {
      "SEAweather": "weather.weather",
      "SEAvisibility": "weather.visibility"
    }
  }
]

```

In the example above, two variables (SEAweather and SEAvisibility) are created. These variables can be used later in text bindings to supply content to the web tile pages.

*Example with “feed” web resource in XML format:*



```
"resources": [
  {
    "url": "http://research.microsoft.com/rss/news.xml",
    "style": "Feed",
    "content": {
      "title": "title"
    }
  }
]
```

#### 2.2.14 icons

The icons section lists any PNG icons that can be used by the web tile within its pages. The set of icons available to the web tile is fixed and all icons are transferred to the band when the web tile is installed. The maximum number of icons that can be defined under the icons section is 8.

For each icon, an internal name is associated with the relative path in the archive where the icon's PNG file is located.

```
"icons" : {
  "SEAIcon" : "icons/SEA.png"
}
```

#### 2.2.15 pages (required)

The pages section provides information about the pages associated with each tile, their layouts, and how their content is populated from the web resources. Each element in the pages array is an object containing the following members:

- **layout** – the name of the layout for the page. The layout name should refer to one of the predefined page layouts. Refer to the [detailed section on predefined page layouts](#) for more information.
- **condition** – if set, then must be set to 'true' for this release.
- **textBindings** – an array of objects corresponding to each page element that contains string content. Each binding object provides the numeric id of the page element based on the predefined layout selected, and the string value that it should contain. The string values contain substitutions from the **content** portions of the **resources** list.
- **iconBindings** – similar to textBindings, but used to select one of the web tile icons in **icons** list for each icon in the page. If there are no icon elements on the page, this member may be omitted. For each icon binding object in the iconBindings array, the members are:
  - **elementId** – the numeric id of the icon layout element based on the predefined layout selected.
  - **conditions** – for this release, this must contain an array with a single “condition” object whose “condition” is true, and whose “icon” member specifies the name of the icon to be used.

```
"pages": [
  {
    "layout": "MSBand_SingleMetricWithSecondary",
    "condition": "true",
    "iconBindings": [
      {
        "elementId": "21",
```

```

        "conditions": [
            {
                "condition": "true",
                "icon": "SEAIcon"
            }
        ]
    },
    "textBindings": [
        {
            "elementId": "11",
            "value": "SEA"
        },
        {
            "elementId": "12",
            "value": "Visibility"
        },
        {
            "elementId": "22",
            "value": "{{SEAvsibility}}"
        }
    ]
}
]

```

### 2.2.16 notifications

Web tiles can include a “notifications” section to describe conditions that should cause the delivery of a dialog notification to the user’s band, along with the content of those notifications. For every data refresh, conditional expressions are evaluated in sequence until one is found to be true and invokes a notification. Each notification declaration includes the following:

- **condition** – (required) the condition to be evaluated to determine whether to trigger the notification. The conditional expression has the syntax “<Operand1> <Operator> <Operand2>” or “true”. Refer to [this section](#) for the complete list of comparison operators supported for condition expressions.
- **title** – (required) the text to use for the notification title. If longer than 20 characters, it will be truncated.
- **body** – the text to use for the notification body. If longer than 20 characters, it will be truncated.

```

"notifications": [
    {
        "condition": "{{SEAweather}} contains \"Cloud\"",
        "title": "SEA is cloudy...",
        "body": "Visibility: {{SEAvsibility}}"
    },
    {
        "condition": "{{SEAweather}} contains \"Rain\"",
        "title": "SEA is rainy...",
        "body": "Visibility: {{SEAvsibility}}"
    }
]

```

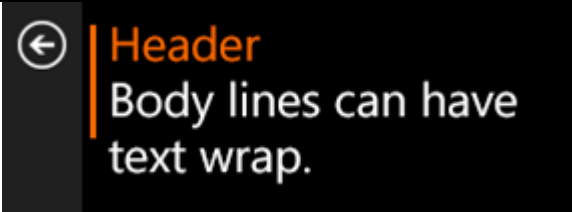
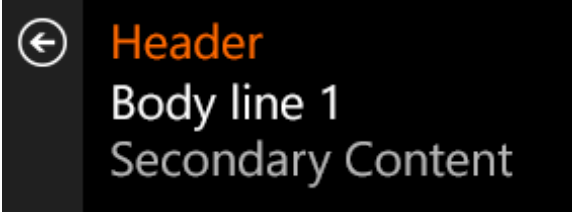
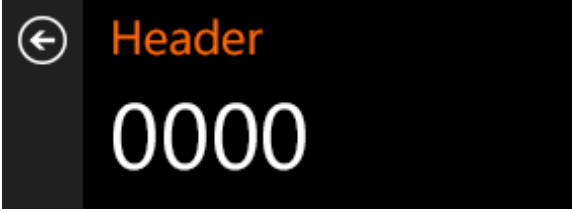
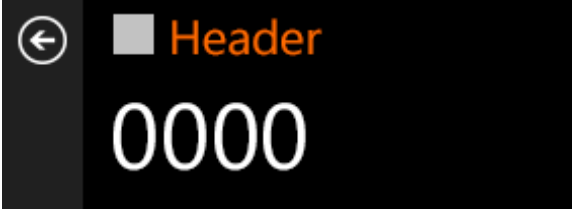
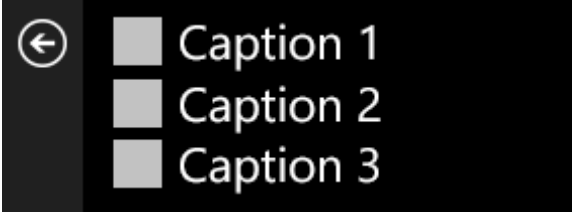
```

    },
    {
      "condition": "{{SEAweather}} contains \"Sunny\"",
      "title": "SEA is sunny!",
      "body": "Visibility: {{SEAvsibility}}"
    }
  ]

```

## 2.3 PREDEFINED LAYOUTS

There are 6 predefined layouts that can be used when creating a web tile.

Layout Name	Preview	Elements
MSBand_ScrollingText		<p><i>ElementId 1:</i> Text string containing a title for the page</p> <p><i>ElementId 2:</i> Wrappable body</p>
MSBand_NoScrollingText		<p><i>ElementId 1:</i> Text string containing a title for the page</p> <p><i>ElementId 2:</i> Primary text</p> <p><i>ElementId 3:</i> Secondary text</p>
MSBand_SingleMetric		<p><i>ElementId 1:</i> Text string containing a title for the page</p> <p><i>ElementId 2:</i> Text string containing digits/characters of a number</p>
MSBand_SingleMetricWithIcon		<p><i>ElementId 11:</i> Small icon that can either take user-selected theme color or custom tile theme color</p> <p><i>ElementId 12:</i> Text string containing a title for the page</p> <p><i>ElementId 21:</i> Text string containing digits/characters of a number</p>
MSBand_MetricsWithIcons		<p><i>ElementId 11:</i> Small icon that can either take user-selected theme color or custom tile theme color</p> <p><i>ElementId 12:</i> Single line text</p> <p><i>ElementId 21:</i> Small icon that can either take user-selected theme color or custom tile theme color</p> <p><i>ElementId 22:</i> Single line text</p>

		<p><i>ElementId 31:</i> Small icon that can either take user-selected theme color or custom tile theme color</p> <p><i>ElementId 32:</i> Single line text</p>
<p><b>MSBand_SingleMetricWithSecondary</b></p>		<p><i>ElementId 11:</i> Text string containing a title for the page</p> <p><i>ElementId 12:</i> Text string containing a secondary header for the page</p> <p><i>ElementId 21:</i> Large icon that will always be set to white</p> <p><i>ElementId 22:</i> Text string containing digits/characters of a number</p>

## 3 WRITING WEB TILES

### 3.1 GETTING STARTED

First we'll want to create a web tile package structure, so let's begin by creating a folder named after your web tile name. Inside your new folder, create another folder called *icons*. Inside the root directory of the web tile folder, create a new text file called *manifest.json* with the following template content. You can rename/edit content as per your web tile.

```
{
  "manifestVersion" : 1,
  "name" : "Web tile name",
  "description" : "App description (100 characters or less)",
  "version" : 1,
  "versionString" : "1",
  "author" : "Author name",
  "contactEmail" : "author@example.com",
  "tileIcon" : {
    "46" : "icons/tileIcon.png"
  },
  "badgeIcon" : {
    "24" : "icons/badgeIcon.png"
  },
  "icons" : {
    "icon1" : "icons/iconIcon.png"
  },
  "refreshIntervalMinutes" : 30,
  "resources" : [
    {
      "url" : "http://mysubdomain.example.com/",
      "style" : "simple",
      "content" : {
        "content1" : "[0].u",
        "content2" : "[1].u",
        "content3" : "[2].u"
      }
    }
  ],
  "pages" : [
    {
      "layout" : "Predefined layouts",
      "condition": "true",
      "iconBindings": [
        {
          "elementId": "1",
          "conditions": [
            {
```

```
        "condition": "true",
        "icon": "icon1"
    }
]
},
"textBindings" : [
    {
        "elementId" : "2",
        "value" : "{{content1}}"
    },
    {
        "elementId" : "3",
        "value" : "{{content2}}"
    },
    {
        "elementId" : "4",
        "value" : "{{content2}}"
    }
]
}
],
"notifications": [
    {
        "condition": "true",
        "title": "Title",
        "body": "Body"
    }
]
}
```

### 3.2 CREATING A “FEED” WEB TILE

This section shows how to create a “feed” web tile that can point to a single web resource that could be either RSS or ATOM data. This type of web tile can create up to 8 pages of the same layout populated with the latest feeds.

1. Define **resources**. The code below creates a variable which will later be used in text bindings to supply content to the web tile pages.

```
"resources": [
  {
    "url": "http://research.microsoft.com/rss/news.xml",
    "style": "Feed",
    "content": {
      "title": "title"
    }
  }
]
```

2. Define **pages**.
  - a. Choose one of the predefined layouts. In this example, the tile uses *MSBand\_ScrollingText* layout.

```
"layout": "MSBand_ScrollingText"
```

- b. Set the page **condition** to be true.
  - c. Define **textBindings**.
    - i. The content for the header (elementId : 1) is a static content “MS Research News”.
    - ii. The content for the body (elementId : 2) is linked to the **content** {{title}} variable.

```
"textBindings": [
  {
    "elementId": "1",
    "value": "MS Research News"
  },
  {
    "elementId": "2",
    "value": "{{title}}"
  }
]
```

Here’s the full code for **pages**.

```
"pages": [
  {
    "layout": "MSBand_ScrollingText",
    "condition": "true",
    "textBindings": [
      {
```

```
    "elementId": "1",  
    "value": "MS Research News"  
  },  
  {  
    "elementId": "2",  
    "value": "{{title}}"  
  }  
]  
}
```



### 3.3 CREATING A “SIMPLE” WEB TILE WITH MULTIPLE RESOURCES AND PAGES

This section shows how to create a “simple” web tile which has three pages with different layouts and extracts content from multiple web resources.



1. List the two image files that will be used in Page 2 and 3 in **icons** section.

```
"icons" : {
  "SEAIcon" : "icons/SEA.png",
  "JFKIcon" : "icons/JFK.png"
}
```

2. Define **resources**. In the code below, two resources are defined.

```
"resources": [
  {
    "url": "http://services.faa.gov/airport/status/SEA?format=json",
    "style": "Simple",
    "content": {
      "SEAweather": "weather.weather",
      "SEAvisibility": "weather.visibility"
    }
  },
  {
    "url": "http://services.faa.gov/airport/status/JFK?format=json",
    "style": "Simple",
    "content": {
      "JFKweather": "weather.weather",
      "JFKvisibility": "weather.visibility"
    }
  }
]
```

3. Define **pages**.
4. Page 1 uses *MSBand\_NoScrollingText* layout.

```
"layout": "MSBand_NoScrollingText"
```

5. Define **textBindings** for Page 1
  - a. The content for the header (elementId: 1) is a static content “Airport Weather”.
  - b. The content for the second line (elementId: 2) is linked to the content {{SEAweather}} variable with “SEA:” static text.
  - c. The content for the third line (elementId: 3) is linked to the content {{JFKweather}} variable with “JFK:” static text.

```
"textBindings": [
  {
    "elementId": "1",
    "value": "Airport Weather"
  },
  {
    "elementId": "2",
    "value": "SEA: {{SEAweather}}"
  },
  {
    "elementId": "3",
    "value": "JFK: {{JFKweather}}"
  }
]
```

6. Page 2 uses *MSBand\_SingleMetricWithSecondary* layout.

```
"layout": "MSBand_SingleMetricWithSecondary"
```

7. Define **textBindings** for Page 2.

- The content for the primary header (elementId: 11) is a static content "SEA".
- The content for the secondary header (elementId: 12) is a static content "Visibility".
- The content for the large body text (elementId: 22) is linked to the content {{SEAvisibility}} variable.

```
"textBindings": [
  {
    "elementId": "11",
    "value": "SEA"
  },
  {
    "elementId": "12",
    "value": "Visibility"
  },
  {
    "elementId": "22",
    "value": "{{SEAvisibility}}"
  }
]
```

8. Define **iconBindings** for Page 2. This page has a single image "SEAIcon" which should have been previously defined in **icons** list.

```
"iconBindings": [
  {
    "elementId": "21",
    "conditions": [
      {
        "condition": true,
```

```

        "icon": "SEAIcon"
      }
    ]
  }
]

```

9. Page 3 also uses *MSBand\_SingleMetricWithSecondary* layout.

```
"layout": "MSBand_SingleMetricWithSecondary"
```

10. Define **textBindings** for Page 3.

- The content for the primary header (elementId: 11) is a static content “JFK”.
- The content for the secondary header (elementId: 12) is a static content “Visibility”.
- The content for the large body text (elementId: 22) is linked to the content {{JFKvisibility}} variable.

```

"textBindings": [
  {
    "elementId": "11",
    "value": "JFK"
  },
  {
    "elementId": "12",
    "value": "Visibility"
  },
  {
    "elementId": "22",
    "value": "{{JFKvisibility}}"
  }
]

```

11. Define **iconBindings** for Page 3. This page has a single image “JFKIcon” which should have been previously defined in **icons** list.

```

"iconBindings": [
  {
    "elementId": "21",
    "conditions": [
      {
        "condition": true,
        "icon": "JFKIcon"
      }
    ]
  }
]

```

Here's the full code for **pages**.

```
"pages": [
  {
    "layout": "MSBand_NoScrollingText",
    "condition": "true",
    "textBindings": [
      {
        "elementId": "1",
        "value": "Aiport Weather"
      },
      {
        "elementId": "2",
        "value": "SEA: {{SEAweather}}"
      },
      {
        "elementId": "3",
        "value": "JFK: {{JFKweather}}"
      }
    ]
  },
  {
    "layout": "MSBand_SingleMetricWithSecondary",
    "condition": "true",
    "iconBindings": [
      {
        "elementId": "21",
        "conditions": [
          {
            "condition": "true",
            "icon": "SEAIcon"
          }
        ]
      }
    ],
    "textBindings": [
      {
        "elementId": "11",
        "value": "SEA"
      },
      {
        "elementId": "12",
        "value": "Visibility"
      },
      {
        "elementId": "22",
        "value": "{{SEAvisibility}}"
      }
    ]
  }
]
```

```

    }
  ]
},
{
  "layout": "MSBand_SingleMetricWithSecondary",
  "condition": "true",
  "iconBindings": [
    {
      "elementId": "21",
      "conditions": [
        {
          "condition": "true",
          "icon": "JFKIcon"
        }
      ]
    }
  ],
  "textBindings": [
    {
      "elementId": "11",
      "value": "JFK"
    },
    {
      "elementId": "12",
      "value": "Visibility"
    },
    {
      "elementId": "22",
      "value": "{{JFKvisibility}}"
    }
  ]
}
]

```