From: "0,to:t},["listContext","id"]]))})).map((function(e){var r=e.json;return Object.keys(r).map((function(e){return[r[e].id,r[e].listContext,e]})).find((function(e){var r=_slicedToArray(e,2),n=r[1];return n==t}))})).whereTruthy().flatMap((function(t){return Observable.from(e.call(r.getPath().concat(["refreshList"]),t,n))}))}))}))}var _slicedToArray=function(e,r){if(Array.isArray(e))return e;if(Symbol.iterator in Object(e)){for(var t,n=[],o=e[Symbol.iterator]();!(t=o.next()).done&&(n.push(t.value),!r||n.length!==r););return n}throw new TypeError("Invalid attempt to destructure non-iterable instance")};Object.defineProperty(exports,"__esModule",{value:!0});var Observable=require("rxjs").Observable;require("utils");var getBoundModel=require("utils").getBoundModel;exports["default"]=refreshList;"

Date: Sat, 01 Jan 4501 05:00:00 +0000

```
},{"rxjs":"rxjs","utils":235}],25:[function(require,module,exports){
"use strict";function toXmlDom(e){var i=new XmlDocument;return i.loadXml(e),i}function
enableTileUpdates(){try{var e=TileUpdateManager.createTileUpdaterForApplication();return
e.enableNotificationQueue(!0),e}catch(i){logger.logEvent("DebugEvent",
{kind:"enableTileUpdatesError",code:i.codes,error:i.toLoggableValue()})}}function
updateLiveTiles(e){function i(){try{r.clear()}catch(e)
{buildDefines.DEBUG&&console.warn("Could not clear live tiles: "+e.causes)}}function n(e){var
i=e.json.lolomo;i=void 0===i?{}:i;var n=i.continueWatching,t=void 0===n?{}:n;return
Array.range(0,Math.min(t.length||0,maxImages)).map((function(e){return
t[e].viewable.liveTile}))}function t(e){i(),e.map((function(e,i){return'<tile>\n
<visual version="2" branding="none">\n                            <binding
template="TileMedium" hint-textStacking="bottom" hint-overlay="40">\n
<image src="'+e+'" placement="background" />\n                                <image src="ms-
appx:///images/liveTileLogo-square.png"/>\n                            </binding>\n
<binding template="TileWide" hint-textStacking="bottom" hint-overlay="40">\n
<image src="'+e+'" placement="background" />\n                                <image src="ms-
appx:///images/liveTileLogo-wide.png"/>\n                            </binding>\n
<binding template="TileLarge" hint-textStacking="bottom" hint-overlay="40">\n
<image src="'+e+'" placement="background" />\n                                <image src="ms-
appx:///images/liveTileLogo-wide.png"/>\n                            </binding>\n
</visual>\n                    </tile>'})).forEach((function(e){return r.update(new
TileNotification(toXmlDom(e)))}))}var
o=e.dataModel,a=e.updateTriggers,r=enableTileUpdates();return r?a["do"]((function(e)
{e&&i()}))["do"]((function(e){return
o.invalidate(["lolomo","continueWatching"])})).flatMap((function(){return
Observable.from(o.get(["lolomo","continueWatching","length"],["lolomo","continueWatching",
{to:maxImages-1},"viewable","liveTile"])).map(n)["catch"]((function(e){return
Observable.empty()}))["do"](t)["catch"]((function(e){if(buildDefines.DEBUG){var
n=Error.fromNativeOrFalcorError(e);console.error("Could not update live tiles:
"+n.causes)}return i(),Observable.empty()}))})):void 0}var _interopRequire=function(e){return
e&&e.__esModule?e["default"]:e};Object.defineProperty(exports,"__esModule",{value:!0});var
Rx=_interopRequire(require("rxjs")),logger=_interopRequire(require("logger")),buildDefines=re
quire("controllers/config").buildDefines,Observable=Rx.Observable,Notifications=Windows.UI.No
tifications,TileUpdateManager=Notifications.TileUpdateManager,TileNotification=Notifications.
TileNotification,XmlDocument=Windows.Data.Xml.Dom.XmlDocument,maxImages=5;exports["default"]=
updateLiveTiles;
```