challenge07.md

Procedure

Please follow the instructions below and use the provided Github template for this week's assignment. You will need to upload your code after challenge 8.

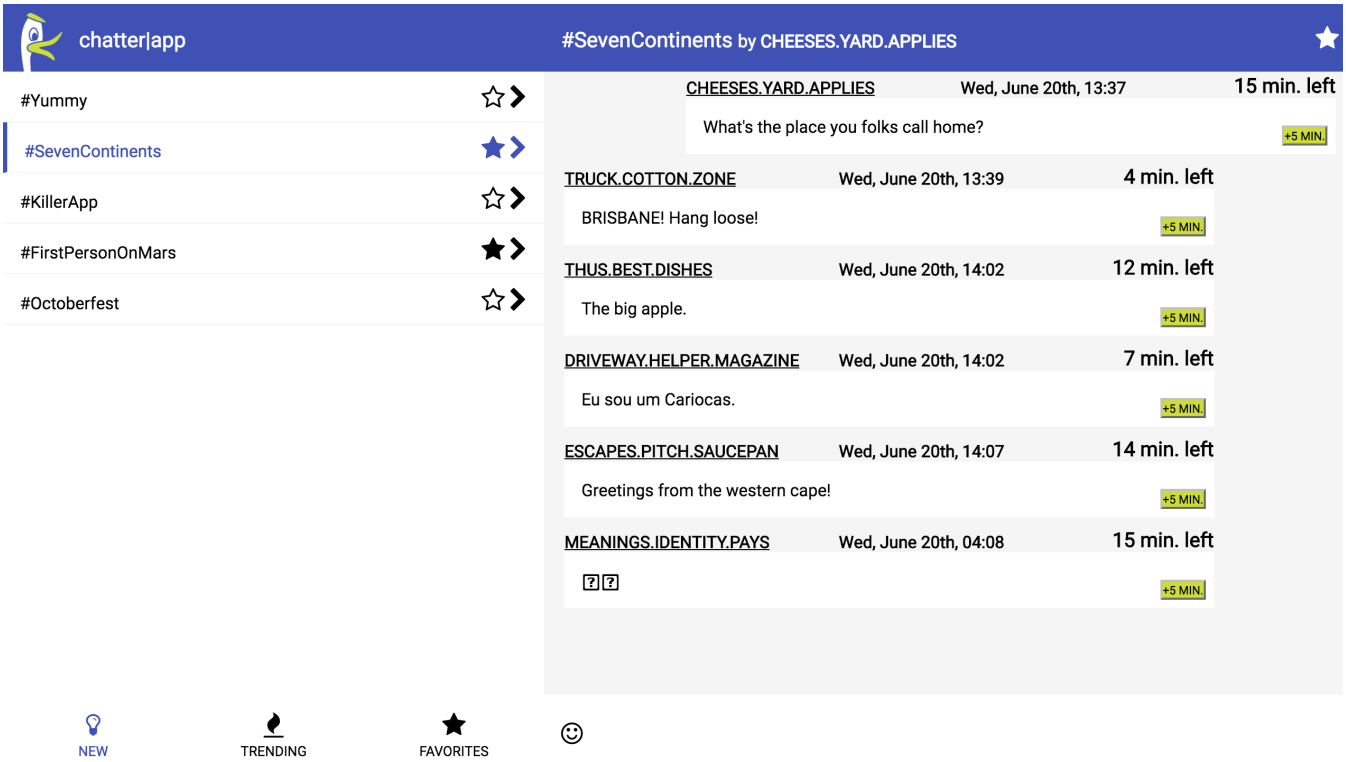
You can download the instructions as pdf here.

There are different exercise types:

- These exercises are important and you should tackle them.
- (*) These asterisk-marked exercises marked are a bit more difficult and thus voluntary. They will improve your skills, tackle them only if you want an extra challenge. However, your app will also work without fulfilling the instruction.

Challenge Goal

This is how your app looks like after **challenge 7**.



It doesn't look very differently to what you already have, because this challenge mainly focuses on interaction.

Graded Criteria

#Icons: Font Awesome is used for all icons, with comparable sizing and functions still working.

- 1 Pt. Channel list icons & star in app bar are replaced by font awesome icons.
- 1 Pt. Clicked stars in channel list turn blue, all others turn to black.
- 1 Pt. Icon size (24px) is comparable to sample solution.
- 1 Pt. Star in app bar switches between filled/unfilled onclick.

#Channel Objects: Channels are treated as an object. An adequate data structure is used. Chatter|app loads from that structure.

- 1 Pt. Channels and their properties are are treated as an object.
- 1 Pt. All Channels are stored in one adequate data structure.

- 2 Pt. Clicking on channel list item, loads channel name, location & star from data structure to write it in right app bar.
- 2 Pt. Location link, loaded from data structure, still works and links to w3w.com .
- 2 Pt. Clicking on star in app bar loads new star from data structure.

whereamivar: currentChannel is global and references the objects' properties.

- 1 Pt. currentChannel is a global variable and is available in console. .
- 1 Pt. A reference of the selected channel's object is stored in currentChannel. Calling currentChannel in console returns its object.

whereamiloc currentLocation is global and location properties are stored.

- 1 Pt. currentLocation is global & is available in console.
- 2 Pt. Location properties are stored in currentLocation. Calling currentLocation in console returns a w3w location and corresponding latitude & longitude data.

Star: currentChannel star toggles when app bar star is toggled.

- 1 Pt. currentChannel star changes accordingly when the app bar's star is toggled.

Your **#syntax** will be graded automatically. Overall, 18 points can be obtained.

Instructions

1. Add awesome icons

- Include the Font Awesome library from BootstrapCDN.
- Replace all stars and chevrons with Font Awesome icons.

Short explanation on inheritance: Notice how the channel stars change to your primary color, in our template app 3F51B5 (INDIGO), when you select them? That is a wonderful example on how inheritance works in CSS. You have written the function switchChannel(channelName) in challenge 6. It now still adds and removes the .selected class. As our star is now an icon and not an image anymore it inherits the .selected element's color property: 3F51B5; So that is why we love inheritance and Font Awesome!

You can also zoom (control + scroll), they are still crisp and sharp (reset zoom by control + 0). That's why we no longer use images.

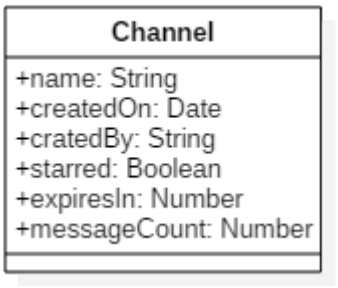
The icons have not the ideal size yet. Material Design requires an icon size of 24px. This is what we need to adjust:

- Change the icon size to 24px (remember: Font Awesome is a font).
- (*) Change the tab bar's icons to Font Awesome icons and fix the size and line breaks (by modifying your css selectors and code).
- (*) Do the same with the smiley button in the chat bar.
- Also replace and resize the star in the right app bar with a filled font-awesome star. Ensure that it's still right-aligned like the image was. It should keep the onclick listener. Fix your code as you don't modify the image's source anymore. In oder to modify the font awesome element's **class** and effectively change the icon, you can use jQuery's $.toggleClass() to toggle the star when clicking on it in the app bar.

2. Treat your channels like objects

- Create a new channels.js in the js folder and embed it in your HTML file. We'll store our channels in there.

- Create a new global variable yummy in the new channels.js and make an object of the yummy channel, using the notation according to the following model:



The yummy channel was created on April 1, 2016 by "minus.plus.yummy", is not starred, will expire in 100 (minutes) and has 999 messages.

- Create variables and objects for the four other channels, taking into account their respective names and if the star is filled from the current prototype and simply duplicate the other data from the yummy channel.
- Modify the arguments for the switchChannel() calls in the channel list's onclick handler. Pass the respective global variable (e.g., yummy) instead of the strings. You don't need the ""! If everything goes well, your switchChannel() acknowledges the received object in the console log:

```
Tuning in to channel                                script.js:10
  Object {name: "#SevenContinents", createdOn: Sat Apr
▶ 02 2016 00:00:00 GMT+0200 (W. Europe Summer Time),
  createdBy: "cheeses.yard.applies", starred: true,
  expiresIn: 60…}
```

- Modify switchChannel() to 'digest' objects. Write the object's properties to your channels app bar (name and location link).

JavaScript knows a conditional operator, which returns one of two different values depending on whether a Boolean is true or false. It tests a boolean condition and returns one of two different values (condition) ? (value if condition is true) : (value if condition is false) An example:

```
true ? 'fa-star' : 'fa-star-o'
will return 'fa-star', whereas

false ? 'fa-star' : 'fa-star-o'
will return 'fa-star-o'.
```

Use this hint to deal with the stars in switchChannel()!

- Let's stick to the app bar. The requirement is to set the star in the app bar (like the channel name and creator) according to our object, when the user clicks on the channel (in the list). If you're successful, the app bar displays the stars according to your objects' definition.

3. Where am I?

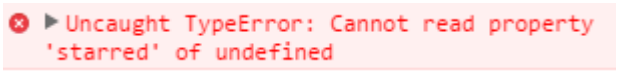
- Create a new global variable currentChannel in your script.js. In switchChannel(), store a reference of the selected channel's object in the currentChannel variable. Test this by calling currentChannel in your console. It does have global scope, if it's available:

```
> currentChannel
<·  Object {name: "#FirstPersonOnMars", createdOn: Wed Sep
  ▶ 28 2016 00:00:00 GMT+0200 (W. Europe Summer Time),
    createdBy: "snipped.atom.grid", starred: true,
    expiresIn: 30003…}
```

- Find out your current geographic longitude and latitude (e.g. using Google, don't do it programmatically yet) and store it in a global variable currentLocation in script.js (don't call the variable location because this already is a global JS object you shouldn't overwrite). Test it with the console. The currentLocation is an object with the three properties longitude, latitude, and what3words. We'll use it for sending messages. Find out the what3words location which is corresponding to your longitude/latitude.

4. Toggle the stars

- Currently, when you press the channel's star, it toggles between star-o and star. Let's also toggle the starred property of your currently selected channel. You can "toggle" Booleans using the not operator. Good that you've already stored the currently active channel. Test in your console if it works.
- (*) You already have the bits and pieces in your code: the conditional operator, the currentChannel, and the jQuery content filter selector. Set the respective star property in your channels list (when the user selects or deselects the star).
- (*) Toggling stars provokes an error when your app is loaded and you have not yet selected a channel. What can you do to fix this?



5. Clean up

- Structure your CSS and write useful comments.
- Check code and syntax. Using W3C Validator helps.

6. Save your code

Do not forget to save your code! You will build upon your work in the next implementation challenge.