ROTTENPOTATOES ENHANCEMENT #3: REMEMBER THE SETTINGS

Now that you have implemented parts 1 and 2, the user can click on the "Movie Title" or "Release Date" headings and see movies sorted by those columns, and can additionally use the checkboxes to restrict the listing to movies with certain ratings only. Moreover, we have preserved RESTfulness, because the URI itself always contains the parameters that will control sorting and filtering.

The last step is to remember these settings. That is, if

- the user has selected any combination of column sorting and restrict-by-rating constraints,

- and then the user clicks to see the details of one of the movies (for example),

then when she clicks "Back to movie list" on the detail page, the movie listing should "remember" the user's sorting and filtering settings from before.

(*Clicking away from the list to see the details of a movie is only one example; the settings should be remembered regardless of what actions the user takes, so that any time she visits the index page, the settings are correctly reinstated.*)

The best way to do the "remembering" will be to use the `session[]` hash. The session is like the `flash[]`, except that once you set something in the `session[]` it is remembered "forever" until you nuke the session with `session.clear` or selectively delete things from it with `session.delete(:some_key)`. That way, in the index method, you can selectively apply the settings from the `session[]` even if the incoming URI doesn't have the appropriate `params[]` set.

To be RESTful, we want to preserve the property that a URI that results in a sorted/filtered view always contains the corresponding sorting/filtering parameters. Therefore, if you find that the incoming URI is lacking the right `params[]` and you're forced to fill them in from the `session[]`, the RESTful thing to do is to `redirect_to` the new URI containing the appropriate parameters. There is an important corner case to keep in mind here, though: if the previous action had placed a message in the `flash[]` to display after a redirect to the movies page, your additional redirect will delete that message and it will never appear, since the `flash[]` only survives across a single redirect. To fix this, use `flash.keep` right before your additional redirect.

Specific Requirements:

- If the user explicitly includes new sorting/filtering settings in `params[]`, the values stored in the `session` should not override them. On the contrary, the new settings should be remembered in the session.

- If a user unchecks all checkboxes, use the settings stored in the `session[]` hash (it doesn't make sense for a user to uncheck all the boxes).

You will submit a text file, containing a single line with the url of your heroku deployment. The url will look like this:

`http://your-app-name.herokuapp.com`

___

Choose Files No file chosen

Check

___

Show Discussion   New Post