

Radio schedule

Module

radio_station

Content type: scheduled_timeslot

field_timeslot - start/end date time, repeating

field_stream - reference to program of type 'Channel/Stream'

field_show - reference to program of type 'Show'

UI

Routes

/schedule

Caching

The cache is built any time a schedule request is made and that week's schedule is not yet cached. Or via one of the available drush commands.

If enabled in the configuration settings, the cache for a given week will be cleared any time a scheduled_timeslot or station_content is saved. As well, cache clear on save will occur during cron when new episodes are generated.

Cron concludes by regenerating the cache for the number of weeks indicated in the configuration settings.

Note during cron the cache is not regenerated when each station_content is created automatically, but rather at the end of the cron run.

The server's crontab is configured to execute Drupal cron at a regular interval.

Cache can be cleared and regenerated upon node save because the URL displayed on the schedule for a program potentially needs to be updated to link to the specific episode rather than the program. Whether this occurs or not is set in the local .env file. On production this is disabled due to performance implications.

All schedule displays (on air, next up, day, week) gets the data from the cache.

As of 4/29/16 this is being done via cron on the production server.. Current settings can be verified in webmin on the server.

This is done and cron is setup via webmin as outlined below. Robert to you to review and close out

```
# 2:15 clear all and regenerate this week
drush @kboo.d7.prod sch-cca;drush @kboo.d7.prod sch-cg

# 2:27 regenerate into a reasonable future
drush @kboo.d7.prod sch-cg --min=1 --max=6

# 2:35 regenerate into a reasonable past
drush @kboo.d7.prod sch-cg --min=-4 --max=-1

# 2:45 regenerate into more of the past
drush @kboo.d7.prod sch-cg --min=-8 --max=-5
```

Drush commands

The following drush commands are available.

```
# List all schedule cache
# doesn't take any parameters
drush schedule-cache-list
drush sch-cl

# Clears all schedule cache
# doesn't take any parameters
drush schedule-cache-clear-all
drush sch-cca

# Clears schedule cache for the given stream
# optional parameters
# stream - defaults to "one"
drush schedule-cache-clear-schedule
drush sch-ccs

# Example: clears stream two cache
drush sch-cc --stream=two

# Clears schedule cache for the given weeks
# optional parameters
# min - defaults to 0
# max - defaults to 0
drush schedule-cache-clear
drush sch-cc

# Example: clears last week, this week, and next week
drush sch-cc --min=-1 --max=1

# Generates schedule cache for the given weeks
# note: if cache already exists for a week then the cache is unaffected
# optional parameters
# min - defaults to 0
# max - defaults to 0
drush schedule-cache-generate
drush sch-cg

# Example: generates last week, this week, and next week
drush sch-cc --min=-1 --max=1

# Clears and regenerates cache for the given weeks
# optional parameters
# min - defaults to 0
# max - defaults to 0
drush schedule-cache-refresh
drush sch-cr

# Example: refresh last week, this week, and next week
drush sch-cr --min=-1 --max=1
```

h4. On now

the class ScheduleComponent defines the following method

```
function getShow($stream)
```

Returns data for the timeslot and, if available, the episode

on-air.coffee implements automated updates

checks every time the minute of the hour ends in 0 or 5

examples: 12:00, 12:05, 12:10, 12:15, ...

JS templating engine used is [Transparency.js|<http://leonidas.github.io/transparency/>]

h4. Tabbed carousels

h5. Up Next

the class ScheduleComponent defines the following method

function getNextShow(\$stream, \$count = 1)

Returns data for the next scheduled episode/timeslot

carousel.coffee implements 'next' and 'prev' requests

h5. Day

the class ScheduleComponent defines the following method

function getDay(\$stream)

Returns data for all of today's scheduled episodes/timeslots

carousel.coffee implements 'next' and 'prev' requests

Week

the class ScheduleComponent defines the following method

function getWeek(\$stream)

Returns data for all of this week's scheduled episodes/timeslots

carousel.coffee implements 'next' and 'prev' requests

Ajax routes

station/episode/\$stream/\$direction/\$timestamp

station/day/\$stream/\$direction/\$timestamp

station/week/\$stream/\$direction/\$timestamp

\$stream = the title or nid of a stream node (i.e. "Stream 1")

\$direction = "prev", "next", "at"

\$timestamp = the timestamp to query against

Episode query uses \$timestamp to find the episode scheduled during that time.

For example, if timestamp converts to "Nov. 5, 2015 3:14pm"

then it would find the episode on Nov. 5, 2015 from 3:00 - 4:00pm

This logic corresponds with the useage below as well.

Day query uses \$timestamp to find the episodes scheduled during that day.
Week query uses \$timestamp to find the episodes scheduled during that week.

Episode Overrides

To override the program for a timeslot, create an episode for an air time that matches the timeslot and set the episode's field_produced_for field to a program other than that associated with the timeslot already. The display for title and showhost are pulled from the selected program.

This has no effect on streamripping. Generated episodes field_produced_for field is automatically set to be the same as the timeslot's program.

For example:

Timeslot A exists for Program 10 at air time Mon 2-3.

I create Program 20.

I create episode for air time Mon 2-3 and associate it with Program 20.

Then schedule for Mon 2-3 would show Program 20 title and link to episode I just created.

When the schedule is loaded it queries scheduled_timeslots nodes. Then it loops through those and for each one queries station_content for an episode set to the same date/time. It then checks the episode's program for override.

FAQ

Question: The schedule cannot be overridden at all if the episode time does not match exactly an existing timeslot? For example, in the example above if I had created the episode from 2:30-3:00 no change in the schedule display would occur? On Air display would also not be different because it goes by the schedule, correct?

Answer: That is correct.

Possible future update: A future update could be done to query all episodes in the date/time range that don't match a timeslot and override the aforementioned results as needed. Note there may be more complicated logic necessary to accommodate multiple episodes results at the same time (e.g. how would the system know which is the override and which is not)

Question: A program must be created in order for an episode to override the display of the schedule? That is, if an episode is created for the same as a timeslot that exists, but it is not associated with any program it will not appear in the schedule?

Answer: Correct.

Possible future update: In the future new logic or new fields could be added to change that.