

# Package ‘myepisodes’

July 2, 2014

**Version** 1.1.1

**Date** 2012-02-17

**Title** MyEpisodes RSS/API functions

**Author** Matt Malin <email@mattmalin.co.uk>

**Maintainer** Matt Malin <email@mattmalin.co.uk>

**Description** Useful functions for accessing MyEpisodes feeds and episode information as well as other tv episode related actions through [www.myepisodes.com](http://www.myepisodes.com)

**Imports** XML

**License** GPL (>= 2)

**Collate** 'myepisodes.R'

**Repository** CRAN

**Date/Publication** 2012-07-03 06:48:30

**NeedsCompilation** no

## R topics documented:

ep_number . . . . .	2
mark_episode_as_acquired . . . . .	2
mark_episode_as_watched . . . . .	3
myepisodes . . . . .	4
myepisodes_feed_url . . . . .	4
shows_from_myepisodes . . . . .	5
shows_from_myepisodes_feed . . . . .	6
show_info_from_xml . . . . .	6
summary_of_shows . . . . .	7
update_episode . . . . .	8
xml_shows_from_myepisodes_feed . . . . .	8

<b>Index</b>	<b>10</b>
--------------	-----------

---

ep_number	<i>Shows episode season/number in format SxNN</i>
-----------	---

---

**Description**

Returns episode from a show info list, of the form e.g. 1x12, 8x04, left padding the episode number to max\_length (default 2) if necessary.

**Usage**

```
ep_number(show_item, max_length = 2)
```

**Arguments**

show_item	a show item list, typically from <a href="#">show_info_from_xml</a>
max_length	the length to pad episode numbers to, usually 2 is standard.

**Value**

character of form NxNN

**Author(s)**

Matt Malin <<email@mattmalin.co.uk>>

**See Also**

[show\\_info\\_from\\_xml](#)

---

mark_episode_as_acquired
--------------------------

*Marks on MyEpisodes if an episode has been acquired (through browser)*

---

**Description**

Given a tv episode (list as from [show\\_info\\_from\\_xml](#)) this marks on MyEpisodes if a show has been acquired. Given the lack of a proper MyEpisodes API and until a better solution is implemented here, it currently loads a browser at the appropriate update link.

**Usage**

```
mark_episode_as_acquired(tv_episode)
```

**Arguments**

tv\_episode      a show\_info list from [show\\_info\\_from\\_xml](#) or similar

**Value**

character of the URL for the myepisodes update link, should also load link in a browser to get the desired behaviour, assuming user is logged in to myepisodes.

**Author(s)**

Matt Malin <<email@mattmalin.co.uk>>

**Examples**

```
## Not run:
mock_feed_url <- file.path(system.file(package = "myepisodes"), "test_data/mock_mylist.xml")
mock_shows <- shows_from_myepisodes_feed(mock_feed_url)
mark_episode_as_acquired(mock_shows[[1]])
## End(Not run)
```

---

mark\_episode\_as\_watched

*Marks on MyEpisodes if an episode has been watched (through browser)*

---

**Description**

Given a tv episode (list as from [show\\_info\\_from\\_xml](#)) this marks on MyEpisodes if a show has been watched. Given the lack of a proper MyEpisodes API and until a better solution is implemented here, it currently loads a browser at the appropriate update link.

**Usage**

```
mark_episode_as_watched(tv_episode)
```

**Arguments**

tv\_episode      a show\_info list from [show\\_info\\_from\\_xml](#) or similar

**Value**

character of the URL for the myepisodes update link, should also load link in a browser to get the desired behaviour, assuming user is logged in to myepisodes.

**Author(s)**

Matt Malin <<email@mattmalin.co.uk>>

**Examples**

```
## Not run:
mock_feed_url <- file.path(system.file(package = "myepisodes"), "test_data/mock_mylist.xml")
mock_shows <- shows_from_myepisodes_feed(mock_feed_url)
mark_episode_as_watched(mock_shows[[1]])
## End(Not run)
```

---

myepisodes	<i>myepisodes functions</i>
------------	-----------------------------

---

**Description**

Various functions to play with MyEpisodes data

**Author(s)**

Matt Malin <<email@mattmalin.co.uk>>

---

myepisodes_feed_url	<i>get the MyEpisodes RSS feed url for a desired feed</i>
---------------------	---

---

**Description**

Returns the custom MyEpisodes RSS feed url for a desired feed.

**Usage**

```
myepisodes_feed_url(uid, pwmd5, feed = "mylist",
  onlyunacquired = TRUE, showignored = FALSE)
```

**Arguments**

uid	myepisodes username
pwmd5	myepisodes MD5 hashed password
feed	the desired feed currently there are: mylist, today, yesterday, tomorrow, unacquired, unwatched, all.
onlyunacquired	TRUE/FALSE - You can use this to decide if you only want the feed to list the episodes you haven't acquired.
showignored	TRUE/FALSE - You can use this to decide if you want the feed to show your ignored shows too.

**Author(s)**

Matt Malin <<email@mattmalin.co.uk>>

**Examples**

```
## Not run: myepisodes_feed_url("foouser", "321654321654321", feed = "mylist", onlyunacquired = TRUE, showignored = TRUE)
```

---

shows\_from\_myepisodes *Get info from shows given desired MyEpisodes feed information*

---

### Description

Given MyEpisodes username and md5 hashed password, along with desired list and other settings, gets show and episode information as a list of show information lists.

### Usage

```
shows_from_myepisodes(uid, pwmd5, feed = "mylist",
    onlyunacquired = TRUE, showignored = FALSE)
```

### Arguments

uid	myepisodes username
pwmd5	myepisodes MD5 hashed password
feed	the desired feed currently there are: mylist, today, yesterday, tomorrow, unacquired, unwatched, all.
onlyunacquired	TRUE/FALSE - You can use this to decide if you only want the feed to list the episodes you haven't acquired.
showignored	TRUE/FALSE - You can use this to decide if you want the feed to show your ignored shows too.

### Value

list of lists of episode information

### Author(s)

Matt Malin <<email@mattmalin.co.uk>>

### See Also

[shows\\_from\\_myepisodes\\_feed](#), [myepisodes\\_feed\\_url](#)

### Examples

```
## Not run: shows_from_myepisodes("foouser", "321654321654321", feed = "mylist", onlyunacquired = TRUE, showignored = FALSE)
```

---

shows\_from\_myepisodes\_feed

*Create list of information for individual shows from MyEpisodes feed*

---

### Description

Given the path to a MyEpisodes feed, extracts the show information for each show, returned as a list.

### Usage

```
shows_from_myepisodes_feed(myepisodes_feed_url)
```

### Arguments

myepisodes\_feed\_url  
path to a MyEpisodes RSS feed

### Value

list, containing show info

### Author(s)

Matt Malin <<email@mattmalin.co.uk>>

### See Also

[show\\_info\\_from\\_xml](#), [myepisodes\\_feed\\_url](#)

### Examples

```
## Not run: shows_from_myepisodes_feed(myepisodes_feed_url("user", "password"))
mock_feed_url <- file.path(system.file(package = "myepisodes"), "test_data/mock_mylist.xml")
shows_from_myepisodes_feed(mock_feed_url)
```

---

show\_info\_from\_xml

*Get tv episode information from XML*

---

### Description

Given an XML show from [xml\\_shows\\_from\\_myepisodes\\_feed](#) in the format of individual items in MyEpisodes feeds, returns a list for a given show containing show\_name, season, ep, ep\_title, and date\_aired.

**Usage**

```
show_info_from_xml(xml_show)
```

**Arguments**

xml\_show           XMLNode of "item" child from MyEpisodes feed

**Value**

list for individual show

**Author(s)**

Matt Malin <<email@mattmalin.co.uk>>

**See Also**

[xml\\_shows\\_from\\_myepisodes\\_feed](#), [myepisodes\\_feed\\_url](#), [shows\\_from\\_myepisodes\\_feed](#)

**Examples**

```
mock_feed_url <- file.path(system.file(package = "myepisodes"), "test_data/mock_mylist.xml")
xml_shows <- xml_shows_from_myepisodes_feed(mock_feed_url)
show_info_from_xml(xml_shows[[1]])
```

---

summary_of_shows	<i>Display summary of all the tv episodes in a list</i>
------------------	---

---

**Description**

Given a list of shows (in form from [show\\_info\\_from\\_xml](#)) will display a summary, displaying each episode of the form "show\_name - SxNN"

**Usage**

```
summary_of_shows(shows)
```

**Arguments**

shows               list of shows as in form from [show\\_info\\_from\\_xml](#)

**Value**

character vector summarising each show

**Author(s)**

Matt Malin <<email@mattmalin.co.uk>>

**Examples**

```
mock_feed_url <- file.path(system.file(package = "myepisodes"), "test_data/mock_mylist.xml")
mock_shows <- shows_from_myepisodes_feed(mock_feed_url)
summary_of_shows(mock_shows)
```

---

update_episode	<i>Marks on MyEpisodes if an episode has been watched/acquired (through browser)</i>
----------------	--

---

**Description**

Given a tv episode (list as from [show\\_info\\_from\\_xml](#)) this marks on MyEpisodes if a show has been watched or acquired. Given the lack of a proper MyEpisodes API and until a better solution is implemented here, it currently loads a browser at the appropriate update link.

**Usage**

```
update_episode(tv_episode, seen = TRUE)
```

**Arguments**

tv_episode	a show_info list from <a href="#">show_info_from_xml</a> or similar
seen	TRUE/FALSE whether to also mark episode as seen, just marks acquired if FALSE

**Value**

character of the URL for the myepisodes update link, should also load link in a browser to get the desired behaviour, assuming user is logged in to myepisodes.

**Author(s)**

Matt Malin <<email@mattmalin.co.uk>>

---

xml_shows_from_myepisodes_feed	<i>Get shows from given MyEpisode feeds (retaining individual XML structure)</i>
--------------------------------	--

---

**Description**

Given a MyEpisodes feed, outputs shows separately as an XMLNodeList for use in [show\\_info\\_from\\_xml](#) or other XML referencing functions.



**Usage**

```
xml_shows_from_myepisodes_feed(myepisodes_feed_url)
```

**Arguments**

```
myepisodes_feed_url  
    path to a MyEpisodes RSS feed
```

**Value**

XMLNodeList with all "item"s from feed, corresponding to episodes

**Author(s)**

Matt Malin <<email@mattmalin.co.uk>>

**See Also**

[show\\_info\\_from\\_xml](#), [myepisodes\\_feed\\_url](#), [shows\\_from\\_myepisodes\\_feed](#)

**Examples**

```
## Not run: xml_shows_from_myepisodes_feed(myepisodes_feed_url("user", "password"))  
mock_feed_url <- file.path(system.file(package = "myepisodes"), "test_data/mock_mylist.xml")  
xml_shows_from_myepisodes_feed(mock_feed_url)
```

# Index

ep\_number, [2](#)

mark\_episode\_as\_acquired, [2](#)  
mark\_episode\_as\_watched, [3](#)  
myepisodes, [4](#)  
myepisodes-package (myepisodes), [4](#)  
myepisodes\_feed\_url, [4](#), [5-7](#), [9](#)

package-myepisodes (myepisodes), [4](#)

show\_info\_from\_xml, [2](#), [3](#), [6](#), [6](#), [7-9](#)  
shows\_from\_myepisodes, [5](#)  
shows\_from\_myepisodes\_feed, [5](#), [6](#), [7](#), [9](#)  
summary\_of\_shows, [7](#)

update\_episode, [8](#)

xml\_shows\_from\_myepisodes\_feed, [6](#), [7](#), [8](#)