

NAME

`ares_set_server_state_callback` – Function for setting a server state callback

SYNOPSIS

```
#include <ares.h>
```

```
typedef void (*ares_server_state_callback)(const char *server_string,
                                           ares_bool_t success,
                                           int flags,
                                           void *data);
```

```
void ares_set_server_state_callback(ares_channel_t *channel,
                                   ares_server_state_callback callback,
                                   void *user_data);
```

DESCRIPTION

The **`ares_set_server_state_callback(3)`** function sets a callback function *callback* in the given ares channel handle *channel* that is invoked whenever a query on the channel completes. This includes both successful and unsuccessful queries (including hard errors and timeouts). The callback function is invoked with a number of parameters describing the query, as follows.

The *server_string* parameter indicates the DNS server that was used for the query, given as a string with the same format returned by **`ares_get_servers_csv(3)`**.

The *success* parameter indicates whether the query succeeded or not. It is set to **`ARES_TRUE`** on success and **`ARES_FALSE`** on failure.

The *flags* parameter is a bitmask of flags describing various aspects of the query (for example whether the query used UDP or TCP). These are described below.

The *data* parameter is a reference to the custom user data *user_data* that was passed to **`ares_set_server_state_callback(3)`** when setting the server state callback.

The server state callback can be used by applications to monitor the state of the DNS servers used by an ares channel. For example, it can be used to track metrics about the numbers and types of queries sent to each server or to detect when a server is uncontactable or unhealthy.

FLAGS**`ARES_SERV_STATE_UDP`**

Indicates that the query was tried over UDP.

`ARES_SERV_STATE_TCP`

Indicates that the query was tried over TCP.

AVAILABILITY

This function was first introduced in c-ares version 1.29.0.

SEE ALSO

`ares_get_servers_csv(3)`