```r
# Load required libraries
library(httr)
library(jsonlite)

# Function to login to API and get token
connect <- function(email, pwd, env) {
  # email: email used to connect to your dnext environment
  # pwd: password
  # env: name of your environment, first part in your dnext URL before the dot
  # Example: ENV.dnext.io -> env = 'ENV'

  # Construct the URL
  url <- paste0("https://api.dnext.io/v1.0/auth/custom-login?org=", env)

  # Create the payload
  payload <- toJSON(list(
    email = email,
    password = pwd,
    organization = env
  ), auto_unbox = TRUE)

  # Set the headers
  headers <- add_headers('Content-Type' = 'application/json')

  # Send the POST request
  response <- POST(url, headers, body = payload)

  # Parse the response and extract the token
  token <- content(response)$token
  return(token)
}

# Call the connect function with your credentials
token <- connect("e.cornette@saftco.ch", "jgyH6564jijliuhçà_mkkjh", "saftco")
token

get_task_status <- function(task_id, token) {
  my_headers <- add_headers(
    Authorization = paste("Bearer", token),
    `Content-Type` = "application/json"
  )

  success <- FALSE
  wait_time <- 2

  while (!success && wait_time < 10) {
    Sys.sleep(wait_time)

    status_res <- GET(paste0('https://api.dnext.io/v1.0/tasks/', task_id), my_headers)

    tryCatch({
      status_json <- fromJSON(content(status_res, as = "text"))
      success <- !is.null(status_json$status) && status_json$status == "SUCCEEDED"
    }, error = function(e) {
      wait_time <- wait_time + 1
    })

    if (!success) {
      wait_time <- wait_time + 1
    }
  }

  return(success)
}
```

```
download_dataset <- function(code, token) {
  my_headers <- add_headers(
    Authorization = paste("Bearer", token),
    `Content-Type` = "application/json"
  )

  dl_url <- paste0('https://api.dnext.io/v1.0/data/datasets/', code, '/download')

  res <- POST(dl_url, my_headers)
  res_json <- fromJSON(content(res, as = "text"))

  if (is.null(res_json$task$id)) {
    stop("Error: Task ID not found in the response.")
  }

  task_id <- res_json$task$id

  data_ready <- get_task_status(task_id, token)

  if (data_ready) {
    if (!is.null(res_json$result$url)) {
      data_url <- res_json$result$url
      data_resp <- GET(data_url)

      df <- read_csv(content(data_resp, as = "raw"), locale = locale(encoding = "UTF-8"),
                     show_col_types = FALSE)
    } else {
      stop("Error: Data URL not found.")
    }
  } else {
    df <- NULL
  }

  return(df)
}

# Usage example
df <- download_dataset("saftco-6a211b50-3b48-4e90-8f86-47e2e2b81909", token)
```