This is another function I wrote to access the MET office API and obtain a 5-day ahead weather forecast:

METDataDownload <- function(stationID, product, key){  
 library("RJSONIO") #Load Library  
 library("plyr")  
 library("dplyr")  
 library("lubridate")  
 connectStr <- paste0("http://datapoint.metoffice.gov.uk/public/data/val/wxfcs/all/json/",stationID,"?res=",product,"&key=",key)  
   
 con <- url(connectStr)  
 data.json <- fromJSON(paste(readLines(con), collapse=""))  
 close(con)  
   
 #Station  
 LocID <- data.json$SiteRep$DV$Location$`i`  
 LocName <- data.json$SiteRep$DV$Location$name  
 Country <- data.json$SiteRep$DV$Location$country  
 Lat <- data.json$SiteRep$DV$Location$lat  
 Lon <- data.json$SiteRep$DV$Location$lon  
 Elev <- data.json$SiteRep$DV$Location$elevation  
   
 Details <- data.frame(LocationID = LocID,  
 LocationName = LocName,  
 Country = Country,  
 Lon = Lon,  
 Lat = Lat,  
 Elevation = Elev)  
 #Parameters  
 param <- do.call("rbind",data.json$SiteRep$Wx$Param)  
   
 #Forecast  
 if(product == "daily"){  
 dates <- unlist(lapply(data.json$SiteRep$DV$Location$Period, function(x){x$value}))  
 DayForecast <- do.call("rbind", lapply(data.json$SiteRep$DV$Location$Period, function(x){x$Rep[[1]]}))  
 NightForecast <- do.call("rbind", lapply(data.json$SiteRep$DV$Location$Period, function(x){x$Rep[[2]]}))  
 colnames(DayForecast)[ncol(DayForecast)] <- "Type"  
 colnames(NightForecast)[ncol(NightForecast)] <- "Type"  
   
 ForecastDF <- plyr::rbind.fill.matrix(DayForecast, NightForecast) %>%  
 as\_tibble() %>%  
 mutate(Date = as.Date(rep(dates, 2))) %>%  
 mutate(Gn = as.numeric(Gn),  
 Hn = as.numeric(Hn),  
 PPd = as.numeric(PPd),  
 S = as.numeric(S),  
 Dm = as.numeric(Dm),  
 FDm = as.numeric(FDm),  
 W = as.numeric(W),  
 U = as.numeric(U),  
 Gm = as.numeric(Gm),  
 Hm = as.numeric(Hm),  
 PPn = as.numeric(PPn),  
 Nm = as.numeric(Nm),  
 FNm = as.numeric(FNm))  
   
   
 } else {  
 dates <- unlist(lapply(data.json$SiteRep$DV$Location$Period, function(x){x$value}))  
 Forecast <- do.call("rbind", lapply(lapply(data.json$SiteRep$DV$Location$Period, function(x){x$Rep}), function(x){do.call("rbind",x)}))  
 colnames(Forecast)[ncol(Forecast)] <- "Hour"  
   
 DateTimes <- seq(ymd\_hms(paste0(as.Date(dates[1])," 00:00:00")),ymd\_hms(paste0(as.Date(dates[length(dates)])," 21:00:00")), "3 hours")  
   
 if(nrow(Forecast) extra\_lines <- length(DateTimes)-nrow(Forecast)  
 for(i in 1:extra\_lines){  
 Forecast <- rbind(rep("0", ncol(Forecast)), Forecast)  
 }  
 }  
   
 ForecastDF <- Forecast %>%  
 as\_tibble() %>%  
 mutate(Hour = DateTimes) %>%  
 filter(D != "0") %>%  
 mutate(F = as.numeric(F),  
 G = as.numeric(G),  
 H = as.numeric(H),  
 Pp = as.numeric(Pp),  
 S = as.numeric(S),  
 T = as.numeric(T),  
 U = as.numeric(U),  
 W = as.numeric(W))  
   
 }  
   
   
 list(Details, param, ForecastDF)  
   
}

The API key can be obtained for free at this link:  
<https://www.metoffice.gov.uk/datapoint/api>

Once we have an API key we can simply insert the station ID and the type of product we want to obtain the forecast. We can select between two products: daily and 3hourly

To obtain the station ID we need to use another query and download an XML with all stations names and ID:

library(xml2)  
  
url = paste0("http://datapoint.metoffice.gov.uk/public/data/val/wxfcs/all/daily/sitelist?key=",key)  
XML\_StationList <- read\_xml(url)  
  
write\_xml(XML\_StationList, "StationList.xml")

This will save an XML, which we can then open with a txt editor (e.g. Notepad++).

The function can be used as follows:

METDataDownload(stationID=3081, product="daily", key)

It will return a list with 3 elements:

1. Station info: Name, ID, Lon, Lat, Elevation
2. Parameter explanation
3. Weather forecast: tibble format

I have not tested it much, so if you find any bug you are welcome to tweak it on GitHub:

<https://github.com/fveronesi/METOfficeForecast>