

Accessing EA Data

The Environment Agency (EA) and Department for Environment Food & Rural Affairs collects a variety of different land, freshwater and marine data to understand the environment in England.

In 2011, the Environment Agency made its environmental monitoring data open to the public. This aimed to increase transparency and accessibility to the public, policymakers and other stakeholders of the state of the environment across England. By doing this it allows organisations working within the same catchments to work from a shared evidence base and a mutual understanding of the state of the environment.

In 2016, Defra (which includes the EA) launched the [Defra Data Services Platform](#) which allows those outside of Defra to search for and access environmental data across all Defra organisations.

This platform has an [App Gallery](#) which lets the public access environmental data through a series of “explorers apps” to explore environmental data. The data that underpins each explorer is also accessible through APIs which when called through a programming language provide access to the raw data that is visualised in these apps.

Data Explorers Apps

Catchment Data Explorer:

Identify Water Framework Directive Classifications for river and groundwater catchments and their sub-catchments. As well as access river basin management plan data.

[England | Catchment Data Explorer](#)

Bathing Waters Explorer:

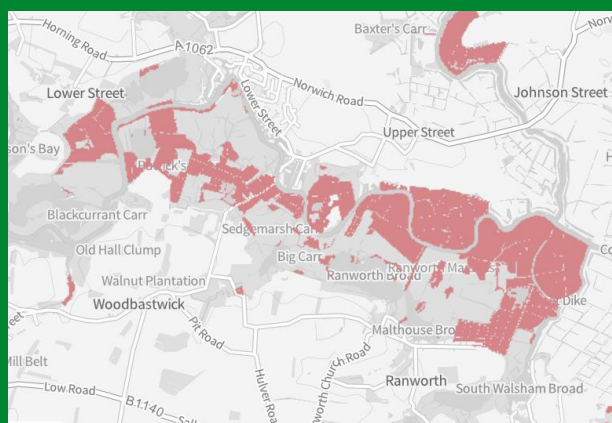
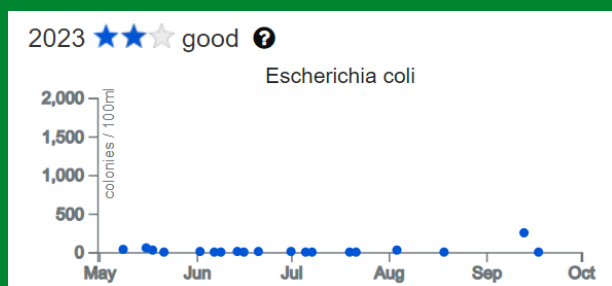
Explore designated bathing waters by name or location.

[Bathing water quality \(data.gov.uk\)](#)

Freshwater Ecology Explorer:

Search, filter and download the Environment Agency’s freshwater ecological open data.

[EA Ecology & Fish Data Explorer](#)



Spatial Data Explorer (MAGIC):

Provides geographic information about the natural environment from across government. This covers rural, urban, coastal and marine environments/habitats across Great Britain.

[MAGIC \(defra.gov.uk\)](https://defra.gov.uk)

Water Quality Data Explorer:

Browse water quality data from the Environment Agency's monitoring sites. Samples are taken at sampling points around England and include coastal or estuarine waters, rivers, lakes, ponds, canals or groundwaters.

[Open WIMS data](#)

Drinking Water & Nitrate Vulnerable Zones:

Browse Drinking Water Safeguard Zones and Nitrate Vulnerable Zones map.

[Drinking Water Safeguard Zones and NVZs](#)

Hydrology Explorer:

Browse data for river flows and water table level readings for monitoring sites across England.

[Hydrology Data Explorer](#)

Rainfall Explorer:

Browse rainfall data from the Environment Agency's rain gauge network.

[Rainfall Data Explorer](#)

Accessing EA Data through an API

If accessing data through the above explorers is not appropriate for your use. Access to the data which is visualised in the above apps are also available through REST APIs.

For information on how to call each API, documentation is available at:

[Defra Data services platform - APIs](#)



Calling an API in Python

```
import requests
import pandas as pd

# Enter your required url endpoints after referring to documentation

root_url = https://environment.data.gov.uk/
endpoint = "...

# Create full URL
url = root_url + endpoint

# Make the API request
response = requests.get(url)
# Check response status and use only valid requests

data = response.json()
df = pd.json_normalize(data, 'items')

# Print the dataframe or transform as wished
print(df.head())
```

Calling an API in R

```
library(jsonlite)
library(httr)

# Enter your required url endpoints after referring to documentation
root_url <- "https://environment.data.gov.uk/"
endpoint <- "...
url <- paste0(root_url, endpoint)

#Load in the api url using the httr and jsonlite packages.
API <- httr::GET(url)
API$status_code #Check if API calling correctly, if 200, it's working

#Convert from json
api <- fromJSON(rawToChar(get$content), flatten=T)

#Create a separate variable
api <- API$items

#Transform dataframe as you wish
```

Example API call of WIMS open data in R

```
# Batch download of all determinants for sites of choice

WIMS_Batch_Download <- function(Sites) {

# Load required libraries
  library(jsonlite)
  library(httr)
  library(tidyverse)

# Define the base URL components and the API limit (max = 9999999)

base_url = "http://environment.data.gov.uk/water-quality/id/sampling-point/"
Deters_list = "/measurements?_"
Limit = "500"

# Function to fetch and process data for selected site

fetch_site_data <- function(sample_ID) {

  url <- paste0(base_url,
                sample_ID,
                Deters_list,
                "limit=", Limit)
  response <- GET(url)

# check the status and handle any errors

if (response$status_code == 200) { api_char <- rawToChar(response$content)

api <- fromJSON(api_char, flatten = TRUE)$items

# Datetime transformations

api %>%
  mutate(
date_time = ymd_hms(sample.sampleDateTime),
Year = year(sample.sampleDateTime)
)
  } else {
  warning(paste("Couldn't fetch data for site:", sample_ID))
  return(NULL)
  }
}

# Apply fetch_site_data function to all sites and combine the results
map_df(Sites, fetch_site_data)

}

# Dataframe for two selected sites.

df <- WIMS_Batch_Download(c("SW-A4070409",
                           "SW-60140265"))

# Carry out other transformations as you wish
```