Cefas/DASSH OGC EDR API Pilot Project

Kate Collingridge & Rebecca Millard | Marine Data Scientists

Together we are working for a sustainable blue future
Pilot project

Aims:

- Test use of Cefas and DASSH OGC EDR APIs to access data programmatically
- Write R functions to query the APIs
- Create an interactive application to visualise data accessed via the APIs
- Provide feedback on how easy the APIs were to use
Querying the API

• Can use swagger to test queries
• Then learn how to construct query URLs.

e.g. https://www.dash.ac.uk/edr/collections/dbossh/instances/DASSHDT00000078/area?coords=POLYGON%28%28-33.31958%20-7.43087%2C-33.31958%2068.91161%2C57.73653%2068.91161%2C57.73653%20-7.43087%2C-33.31958%20-7.43087%29%29
Functions

- R code to return usable data given basic inputs
- Used in Rshiny application
- E.g:
  - To construct a query url
  - To take a shape drawn on a map and convert it into a polygon query to return data within that shape.
  - To extract the spatial information for a dataset to plot it onto a map

```r
retrieve_Cefas_OGC_data <- function(collectionId, query_type = "whole dataset", polygon_url) {
  # define API link
  url <- "https://data-api.cefas.co.uk/ogc/

  # create collection url and retrieve info
  url_collection <- paste0(url, "collections/", collectionId)
  dat_collection <- http::GET(url_collection)

  # get content and convert from json
  content_collection <- http::content(dat_collection, as = "text")
  content_collection <- jsonlite::fromJSON(content_collection)

  # get extent of spatial data (bounding box)
  bbox <- content_collection$extent$spatial$boundingBox

  if(query_type == "Whole dataset") {
    # make the bounding box a polygon query url for collection
    # create query URL with bounding box coordinates
    # note: list coordinates from the bottom left in clockwise order, repeat bottom left at the end
    dat_collection_all <- http::GET(paste0(url_collection,
      "/area?coords=POLYGON",
      RCurl::curlEscape(paste0(
  } else if(query_type == "Draw polygon") {

    # get content and convert from json
    content_collection_all <- http::content(dat_collection_all, as = "text")
    content_collection_all <- jsonlite::fromJSON(content_collection_all)

    # return features from data
    return(content_collection_all$features)
  }
}
```
R Shiny App

- **Shiny (rstudio.com)**
- App allows users to query APIs and visualise data without needing technical expertise.
- Hosted on Cefas’ rstudioconnect infrastructure.
Lessons learned

- Differing Implementation – made writing standardised functions difficult

<table>
<thead>
<tr>
<th>Cefas</th>
<th>DASSH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection1</td>
<td>Collection-dbossh</td>
</tr>
<tr>
<td>data</td>
<td>Instance-survey1</td>
</tr>
<tr>
<td>Collection2</td>
<td>data</td>
</tr>
<tr>
<td>data</td>
<td>Instance-survey2</td>
</tr>
<tr>
<td>Collection3</td>
<td>data</td>
</tr>
<tr>
<td>data</td>
<td>Instance-survey3</td>
</tr>
</tbody>
</table>
Lessons learned

- Timeouts – data not returned, why?
- Data volume – DASSH has a large amount of data so most queries take a long time or timeout.
Lessons learned

• Blind querying – difficult to use an API if you don’t already know what data it serves!
Lessons learned

- API not setup to return complete datasets easily

"Return the entire dataset in collection ID x" is not possible

... but there are workarounds!

"Get the spatial extent of dataset in collection ID x and use this to return the entire dataset in collection ID x" is possible
Thank you for listening

Follow @CefasGovUK

Visit cefas.co.uk

Subscribe to our newsletters

@ data.manager@cefas.co.uk (soon to be @cefas.gov.uk)

Together we are working for a sustainable blue future