# Common Evidence Explorer Plugin

2021-08-09

## Contents



The common evidence explorer is a shiny application and module that can be used within other shiny applications that use OMOP standard vocabularies.

### Launching the explorer

Using the explorer works with either a database cem connector backend or a hosted url solution. The simplest solution is to use an API as follows. This will launch the shiny app:

```
baseUrl <- "https://cem.ohdsi.org/api/v1/"
launchCeExplorer(apiUrl = baseUrl)
```

Alternatively, if database credentials are known the app can launch as follows:

```
connectionDetails <- DatabaseConnector::createConnectionDetails("postgres", ...)
launchCeExplorer(connectionDetails = connectionDetails, cemSchema = "cem", sourceSchema = "cem_v3_sourc
```

Consult your organisations administrator for details on connecting in database mode. Unless you are serving a large number of requests, this mode is not required.

### Using the explorer shiny module

This approach allows you to import the Shiny module in to your shiny application of choice.

For example:

```
serverFunction <- function(input, output, session) {
  backend <- CemConnector::CemWebApiBackend$new(apiUrl = "https://cem.ohdsi.org/api/v1/")
  # Define a reactive that returns a dataframe with standard ingredient concepts
  ingredientConceptInput <- shiny::reactive({ tibble::tibble(conceptId = c(21604296), includeDescendants
  # Define a reactive that returns a dataframe with standard condtion concepts
  conditionConceptInput <- shiny::reactive({ tibble::tibble(conceptId = c(4149320), includeDescendants =
  # Define a reactive that returns an integer that for looking up higher levels in the heirarchy
  siblingLookupLevelsInput <- shiny::reactive({ 1 })

  # Call the explorer module
  ceModuleServer <- ceExplorerModule("explorer", # This ID should be unqiue and match the call to ceExp
```

```
                                      backend,
                                      ingredientConceptInput = ingredientConceptInput,
                                      conditionConceptInput = conditionConceptInput,
                                      siblingLookupLevelsInput = siblingLookupLevelsInput)
}

uiFunction <- function () {
  explorerTable <- ceExplorerModuleUi("exporer") # The id for the server module and ui should be the sa
  shiny::fluidPage(explorerTable)
}

shiny::shinyApp(uiFunction, serverFunction)
```

The reactive objects for ingredientConceptInput and conditionConceptInput must return data.frames that are either empty or contain the column names `conceptId`, `includeDescendants` and `isExcluded`.