

# Package ‘iSEEindex’

April 2, 2025

**Title** iSEE extension for a landing page to a custom collection of data sets

**Version** 1.5.0

**Date** 2024-10-09

**Description** This package provides an interface to any collection of data sets within a single iSEE web-application. The main functionality of this package is to define a custom landing page allowing app maintainers to list a custom collection of data sets that users can selected from and directly load objects into an iSEE web-application.

**License** Artistic-2.0

**URL** <https://github.com/iSEE/iSEEindex>

**BugReports** <https://support.bioconductor.org/t/iSEEindex>

**biocViews** Software, Infrastructure

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.2

**Depends** SummarizedExperiment, SingleCellExperiment

**Imports** BiocFileCache, DT, iSEE, methods, paws.storage, rintrojs, shiny, shinydashboard, shinyjs, stringr, urltools, utils

**Suggests** BiocStyle, covr, knitr, RefManageR, rmarkdown, markdown, scRNAseq, sessioninfo, testthat (>= 3.0.0), yaml

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**git\_url** <https://git.bioconductor.org/packages/iSEEindex>

**git\_branch** devel

**git\_last\_commit** 7211e8a

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.21

**Date/Publication** 2025-04-02

**Author** Kevin Rue-Albrecht [aut, cre] (ORCID: <https://orcid.org/0000-0003-3899-3872>),  
 Thomas Sandmann [ctb] (ORCID: <https://orcid.org/0000-0002-6601-8890>),  
 Federico Marini [aut] (ORCID: <https://orcid.org/0000-0003-3252-7758>),  
 Denali Therapeutics [fnd]

**Maintainer** Kevin Rue-Albrecht <kevinrue67@gmail.com>

## Contents

|  |           |
|--|-----------|
| iSEEindex-package . . . . .                | 2         |
| iSEEindex . . . . .                        | 3         |
| iSEEindexHttpsResource-class . . . . .     | 7         |
| iSEEindexLocalhostResource-class . . . . . | 8         |
| iSEEindexRcallResource-class . . . . .     | 10        |
| iSEEindexResource-class . . . . .          | 11        |
| iSEEindexResource-generics . . . . .       | 12        |
| iSEEindexRunrResource-class . . . . .      | 13        |
| iSEEindexS3Resource-class . . . . .        | 14        |
| <b>Index</b>                               | <b>17</b> |

---

|                   |   |
|-------------------|---|
| iSEEindex-package | <i>iSEEindex: iSEE extension for a landing page to a custom collection of data sets</i> |
|-------------------|---|

---

## Description

This package provides an interface to any collection of data sets within a single iSEE web-application. The main functionality of this package is to define a custom landing page allowing app maintainers to list a custom collection of data sets that users can select from and directly load objects into an iSEE web-application.

## Author(s)

**Maintainer:** Kevin Rue-Albrecht <kevinrue67@gmail.com> (ORCID)

Authors:

- Federico Marini <marinif@uni-mainz.de> (ORCID)

Other contributors:

- Thomas Sandmann <tomsing1@gmail.com> (ORCID) [contributor]
- Denali Therapeutics [funder]

**See Also**

Useful links:

- <https://github.com/iSEE/iSEEindex>
- Report bugs at <https://support.bioconductor.org/t/iSEEindex>

---

iSEEindex

*iSEEindex App*


---

**Description**

Generate an **iSEE** app that includes a landing page enabling users to choose from a custom set of data sets and initial configuration states prepared by the app maintainer.

**Usage**

```
iSEEindex(
  bfc,
  FUN.datasets,
  FUN.initial = NULL,
  default.add = TRUE,
  default.position = c("first", "last"),
  app.title = NULL,
  body.header = NULL,
  body.footer = NULL
)
```

**Arguments**

|                  |   |
|------------------|---|
| bfc              | An <a href="#">BiocFileCache()</a> object.  |
| FUN.datasets     | A function that returns a list of metadata for available data sets.   |
| FUN.initial      | A function that returns a list of metadata for available initial configuration states.  |
| default.add      | Logical scalar indicating whether a default initial configuration should be added as a choice in the Shiny <code>selectizeInput()</code> . See <a href="#">iSEEindex()</a> .  |
| default.position | Character scalar indicating whether the default initial configuration should be added as the "first" or "last" option in the Shiny <code>selectizeInput()</code> .  |
| app.title        | Character string to specify the desired title to be displayed in the main window of the dashboard. Defaults to NULL, which displays some info on the versions of the <code>iSEEindex</code> and <code>iSEE</code> packages. |
| body.header      | UI element to display <i>above</i> the main landing page body.  |
| body.footer      | UI element to display <i>below</i> the main landing page body.  |

**Value**

An `iSEE::iSEE()` app with a custom landing page using a `BiocFileCache()` to cache a selection of data sets.

**Data Sets**

The function passed to the argument `FUN.datasets` must return a `list` that contains metadata about the available data sets.

For each data set, required metadata are:

**id** A unique identifier for the data set.

**title** A short human-readable title for the data set, displayed in the 'Info' panel when the data set is selected.

**uri** A Uniform Resource Identifier (URI) that indicates the location of the data file that contains the data set.

**description** A more detailed description of the data set, displayed in the 'Info' panel when the data set is selected.

Example:

```
list(
  list(
    id = "dataset01",
    title = "Dataset 01",
    uri = "https://example.com/1.rds",
    description = "My first data set."
  ),
  list(
    id = "dataset02",
    title = "Dataset 02",
    uri = "https://example.com/1.rds",
    description = "My second data set."
  )
)
```

The individual sub-lists may also contain optional named metadata specific to individual `iSEEindexResource` classes (refer to the help page of those classes for details).

**Important:** The `id` value is used to identify the data set file in the `BiocFileCache`. Thus, we recommend using a dedicated `BiocFileCache()` for the app, using the `BiocFileCache(cache)` argument to specify an on-disk location (directory path) for the dedicated cache.

**Initial Configurations**

The function passed to the argument `FUN.initial` must return a `list` that contains metadata about the available initial configurations, or `NULL` in the absence of any custom initial configuration (default settings will be applied to all data sets.).

For each initial configuration, required metadata are:

- id** A unique identifier for the initial configuration.
- title** A short human-readable title for the initial configuration, representing the initial configuration in the 'Initial settings' dropdown menu.
- uri** A Uniform Resource Identifier (URI) that indicates the location of the R script that contains the initial configuration.
- description** A more detailed description of the initial configuration, displayed in the 'Configure and launch' panel when the initial configuration is selected.

For each initial configuration, optional metadata are:

- datasets** A series of data set identifiers for which the configuration should be made available. If missing, the configuration will be available for all data sets.

Example:

```
list(
  list(
    id = "config01",
    datasets = c("dataset01")
    title = "Configuration 01",
    uri = "https://example.com/1.R",
    description = "My first configuration."
  ),
  list(
    id = "config02",
    title = "Configuration 02",
    uri = "https://example.com/2.R",
    description = "My second configuration."
  )
)
```

The individual sub-lists may also contain additional optional named metadata specific to individual [iSEEindexResource](#) classes (refer to the help page of those classes for details).

### Author(s)

Kevin Rue-Albrecht

### Examples

```
library("BiocFileCache")
bfc <- BiocFileCache(cache = tempdir())

dataset_fun <- function() {
  x <- yaml::read_yaml(system.file(package = "iSEEindex", "example.yaml"))
  x$datasets
}

initial_fun <- function() {
```

```

    x <- yaml::read_yaml(system.file(package = "iSEEindex", "example.yaml"))
    x$initial
  }

app <- iSEEindex(bfc, dataset_fun, initial_fun)

if (interactive()) {
  shiny::runApp(app, port = 1234)
}

## Alternatively, with the example based on using runr calls

dataset_fun_tonsils <- function() {
  x <- yaml::read_yaml(
    system.file("tonsils_example", "tonsil_package.yaml", package = "iSEEindex")
  )
  x$datasets
}

initial_fun_tonsils <- function() {
  x <- yaml::read_yaml(
    system.file("tonsils_example", "tonsil_package.yaml", package = "iSEEindex")
  )
  x$initial
}

library("shiny")
header_tonsils <- fluidRow(
  shinydashboard::box(
    width = 12, collapsible = TRUE, collapsed = TRUE,
    title = "How to explore the Tonsil Atlas datasets",
    includeMarkdown(
      system.file("tonsils_example", "header_tonsils.md", package = "iSEEindex")
    )
  )
)

footer_tonsils <- fluidRow(
  shinydashboard::box(
    width = 12,
    includeMarkdown(
      system.file("tonsils_example", "footer_tonsils.md", package = "iSEEindex")
    )
  )
)

app_tonsils <- iSEEindex(bfc,
  dataset_fun_tonsils,
  initial_fun_tonsils,
  default.add = TRUE,
  default.position = "last",
  app.title = "iSEE loves the Tonsil Data Atlas",
  body.header = header_tonsils,
  body.footer = footer_tonsils)

```

```
if (interactive()) {
  shiny::runApp(app_tonsils, port = 5678)
}

## This example shows that it is possible to mix different types of resources
## Some provide the path, some directly the object

dataset_fun_mix <- function() {
  x <- yaml::read_yaml(
    system.file("mixed_resources.yml", package = "iSEEindex")
  )
  x$datasets
}
initial_fun_mix <- function() {
  x <- yaml::read_yaml(
    system.file("mixed_resources.yml", package = "iSEEindex")
  )
  x$initial
}

app_mixed <- iSEEindex(bfc,
  dataset_fun_mix,
  initial_fun_mix,
  default.add = TRUE,
  default.position = "last",
  app.title = "iSEE loves multiple resource types")

if (interactive()) {
  shiny::runApp(app_mixed, port = 4242)
}
```

---

iSEEindexHttpsResource-class

*The iSEEindexHttpsResource class*

---

## Description

The `iSEEindexHttpsResource` class represents a resource accessible through an HTTPS link. A URI for this type of resource uses the prefix “https://”.

## Usage

```
iSEEindexHttpsResource(x)
```

## Arguments

x                    List of metadata. See Details.

## Details

Required metadata:

**uri** Character scalar. URI of the resource.

## Value

The constructor function `iSEEindexHttpsResource()` returns an object of class `iSEEindexHttpsResource`.

## Slot overview

This class inherits all slots from its parent class `iSEEindexResource`.

## Supported methods

In the following code snippets, `x` is an instance of a `iSEEindexHttpsResource` class. Refer to the documentation for each method for more details on the remaining arguments.

- `precache(x, bfc, id, ...)` caches the resource located at the given URI using **BiocFile-Cache** and returns the file path to the cached file.

## Author(s)

Kevin Rue-Albrecht

## Examples

```
iSEEindexHttpsResource(list(uri = "https://example.com"))
```

---

`iSEEindexLocalhostResource-class`

*The iSEEindexLocalhostResource class*

---

## Description

The `iSEEindexLocalhostResource` class represents a resource accessible through a local filepath. A URI for this type of resource uses the prefix “localhost://”.

## Usage

```
iSEEindexLocalhostResource(x)
```

## Arguments

`x` List of metadata. See Details.



**Details**

Required metadata:

**uri** Character scalar. URI of the resource.

**Value**

The constructor function `iSEEindexLocalhostResource()` returns an object of object of class `iSEEindexLocalhostResource`.

**Slot overview**

This class inherits all slots from its parent class `iSEEindexResource`.

**Supported methods**

In the following code snippets, `x` is an instance of a `iSEEindexLocalhostResource` class. Refer to the documentation for each method for more details on the remaining arguments.

- `precache(x, ...)` trims the `localhost://` prefix, and caches a copy of the resource located at the resulting file path using `BiocFileCache`, before returning the file path to the cached file.

**Absolute and relative paths**

Absolute and relative paths are both supported.

Absolute paths require an additional `/` (forward slash) following the double forward slash `//` separating the scheme component of the URI.

For instance:

- `localhost://path/to/file` refers to the relative path `path/to/file` (relative to the working directory when the Shiny application is launched).
- `localhost:///path/to/file` refers to the absolute path `/path/to/file`.

**Author(s)**

Kevin Rue-Albrecht

**Examples**

```
iSEEindexLocalhostResource(list(uri = "localhost:///example/absolute/path"))  
iSEEindexLocalhostResource(list(uri = "localhost://example/relative/path"))
```

---

iSEEindexRcallResource-class

*The iSEEindexRcallResource class*

---

### Description

The `iSEEindexRcallResource` class represents a resource accessible through the result of an R call. A URI for this type of resource uses the prefix “`rcall://`”.

### Usage

```
iSEEindexRcallResource(x)
```

### Arguments

`x` List of metadata. See Details.

### Details

Required metadata:

**uri** Character scalar. R call which, once evaluated, produces a character scalar that is the URI of the resource.

### Value

The constructor function `iSEEindexRcallResource()` returns an object of object of class `iSEEindexRcallResource`.

### URI format

The URI must contain valid R code, once the prefix `rcall://` is removed. The code must return the path to an existing file on the local filesystem.

For instance:

```
rcall://system.file(package='iSEEindex', 'ReprocessedAllenData_config_01.R')
```

### Slot overview

This class inherits all slots from its parent class [iSEEindexResource](#).

### Supported methods

In the following code snippets, `x` is an instance of a [iSEEindexRcallResource](#) class. Refer to the documentation for each method for more details on the remaining arguments.

- `precache(x, ...)` trims the `rcall://` prefix, evaluates the remainder of the URI as R code, and caches a copy of the resource located at the resulting file path using **BiocFileCache**, before returning the file path to the cached file.

**Author(s)**

Kevin Rue-Albrecht

**Examples**

```
iSEEindexRcallResource(list(  
  uri = "rcall://system.file(package='iSEEindex', 'ReprocessedAllenData_config_01.R')"  
))
```

---

iSEEindexResource-class

*The iSEEindexResource class*

---

**Description**

The iSEEindexResource class is a virtual class from which classes of supported resource must be derived.

**Usage**

```
## S4 method for signature 'iSEEindexResource'  
show(object)  
  
## S4 method for signature 'iSEEindexResource'  
precache(x, bfc, id, ...)
```

**Arguments**

|        |  |
|--------|--|
| object | An iSEEindexResource object.                           |
| x      | An <a href="#">iSEEindexResource</a> object.           |
| bfc    | A <a href="#">BiocFileCache()</a> object.              |
| id     | A data set identifier as a character scalar.           |
| ...    | additional arguments passed to and from other methods. |

**Value**

show() returns NULL after displaying a summary of the object.  
precache() throws an error if no method is found for the derived class.

**Slot overview**

- uri, a character scalar specifying the URI of a resource.

### Supported methods

In the following code snippets, `x` is an instance of a `iSEEindexResource` class. Refer to the documentation for each method for more details on the remaining arguments.

- `precache(x, bfc, id, ...)` throws an error, encouraging users to develop a method for derived classes that are not supported yet.

### Author(s)

Kevin Rue-Albrecht

### Examples

```
showClass("iSEEindexResource")
```

---

iSEEindexResource-generics

*Generics for iSEEindexResources Objects*

---

### Description

An overview of the generics for `iSEEindexResources` objects.

### Usage

```
precache(x, bfc, id, ...)
```

### Arguments

|                  |  |
|------------------|--|
| <code>x</code>   | An <code>iSEEindexResource</code> object.              |
| <code>bfc</code> | A <code>BiocFileCache()</code> object.                 |
| <code>id</code>  | A data set identifier as a character scalar.           |
| <code>...</code> | additional arguments passed to and from other methods. |

### Value

`precache()` returns the file path to the cached copy of a resource fetched from a given URI.

### Preparing and caching resources

`precache(x, bfc, id, ...)` retrieves and caches a resource from an URI, caches it, and returns the path to the cached file.

### Author(s)

Kevin Rue-Albrecht

**Examples**

```
library(BiocFileCache)
bfc <- BiocFileCache(cache = tempdir())

x <- iSEEindexRcallResource(list(
  uri = "rcall://system.file(package='iSEEindex','ReprocessedAllenData_config_01.R')")
))
precache(x, bfc, "ID0")
```

---

iSEEindexRunrResource-class

*The iSEEindexRunrResource class*


---

**Description**

The iSEEindexRunrResource class represents an SE object, obtained directly through an R call.

**Usage**

```
iSEEindexRunrResource(x)
```

**Arguments**

**x** List of metadata. See Details.

**Details**

A URI for this type of resource uses the prefix “runr://”.

Required metadata:

**uri** Character scalar. URI of the resource.

**Value**

The constructor function iSEEindexRunrResource() returns an object of object of class iSEEindexRunrResource.

**Supported methods**

In the following code snippets, x is an instance of a [iSEEindexRunrResource](#) class. Refer to the documentation for each method for more details on the remaining arguments.

- `precache(x, bfc, id, ...)` caches the resource located at the given URI using **BiocFileCache** and returns the file path to the cached file.

**Examples**

```
iSEEindexRunrResource(list(
  uri = "runr://HCA_TonsilData::HCA_TonsilData(assayType = 'RNA', cellType = 'epithelial')")
))
```

---

iSEEindexS3Resource-class

*The iSEEindexS3Resource class*

---

## Description

The iSEEindexS3Resource class represents a cloud storage resource accessible via the [paws.storage](#) R package. A URI for this type of resource uses the prefix “s3://”.

## Usage

```
iSEEindexS3Resource(x)
```

## Arguments

x                      List of metadata. See Details.

## Details

Required metadata:

**uri** Character scalar. URI of the resource.

## Value

The constructor function `iSEEindexS3Resource()` returns an object of object of class `iSEEindexS3Resource`.

## Slot overview

This class inherits all slots from its parent class [iSEEindexResource](#).

Furthermore, this class defines the additional slot(s) below:

**region** AWS region.

## Supported methods

In the following code snippets, `x` is an instance of a [iSEEindexS3Resource](#) class. Refer to the documentation for each method for more details on the remaining arguments.

- `precache(x, ..., temp_dir = tempdir())` trims the `s3://` prefix, parses information encoded in the remainder of the URI, downloads the resource from AWS S3 using that information, and caches a copy of the resource located at the resulting file path using **BiocFileCache**, before returning the file path to the cached file.

**URI format**

The URI must correspond to an existing file in an AWS S3 compatible cloud storage system.

For instance:

```
s3://bucket/prefix/index.rds
```

For details about authentication, see section “AWS Credentials” below.

**Pre-caching**

Additional arguments to the `precache(x, ..., temp_dir = tempdir())`:

`temp_dir` Scalar character, the directory to store the downloaded file in before it is handed over to **BiocFileCache**. This directory will be created recursively if it doesn't already exist.

**AWS Credentials and region settings**

For detailed information, please consult the [paws R package documentation](#).

Currently, you must have the [AWS Command Line Interface](#) installed to use AWS SSO with **paws.storage**.

A default AWS region can be set in the file `~/.aws/config`. For instance:

```
[default]
region=eu-west-2
```

Optionally, a field named `region` can be added in the list of resource metadata to set the AWS S3 region for each individual resource, e.g.

```
- id: ID1
  title: ReprocessedAllenData
  uri: s3://example/ReprocessedAllenData.rds
  description: |
    Reprocessed Allen Data.
  region: eu-west-2
```

Regions set in individual resource metadata override the default AWS region set in `~/.aws/config` (if any). The region metadata does not need to be set for resources that should use the default region, and resource classes that do not require region information.

If a default region is NOT set in `~/.aws/config`, then the region MUST be set in the metadata.

Credentials for all services can be set in the AWS shared credentials file `~/.aws/credentials`. For instance:

```
[default]
aws_access_key_id=your AWS access key
aws_secret_access_key=your AWS secret key
```

**Author(s)**

Thomas Sandmann, Kevin Rue-Albrecht

**Examples**

```
# Without region metadata
metadata <- list(uri = "s3://example/path/to/bucket")
x <- iSEEindexS3Resource(metadata)
str(x)

# With region metadata
# NOTE: The @region slot is set to NA pending bugfix (see above).
metadata <- list(uri = "s3://example/path/to/bucket", region = "eu-west-2")
x <- iSEEindexS3Resource(metadata)
str(x)
```



# Index

## \* **internal**

- iSEEindex-package, [2](#)
- BiocFileCache(), [3](#), [4](#), [11](#), [12](#)
- iSEE::iSEE(), [4](#)
- iSEEindex, [3](#)
- iSEEindex(), [3](#)
- iSEEindex-package, [2](#)
- iSEEindexHttpsResource, [8](#)
- iSEEindexHttpsResource
  - (iSEEindexHttpsResource-class), [7](#)
- iSEEindexHttpsResource-class, [7](#)
- iSEEindexLocalhostResource, [9](#)
- iSEEindexLocalhostResource
  - (iSEEindexLocalhostResource-class), [8](#)
- iSEEindexLocalhostResource-class, [8](#)
- iSEEindexRcallResource, [10](#)
- iSEEindexRcallResource
  - (iSEEindexRcallResource-class), [10](#)
- iSEEindexRcallResource-class, [10](#)
- iSEEindexResource, [4](#), [5](#), [8–12](#), [14](#)
- iSEEindexResource-class, [11](#)
- iSEEindexResource-generics, [12](#)
- iSEEindexRunrResource, [13](#)
- iSEEindexRunrResource
  - (iSEEindexRunrResource-class), [13](#)
- iSEEindexRunrResource-class, [13](#)
- iSEEindexS3Resource, [14](#)
- iSEEindexS3Resource
  - (iSEEindexS3Resource-class), [14](#)
- iSEEindexS3Resource-class, [14](#)
- precache, [8–10](#), [12–15](#)
- precache (iSEEindexResource-generics), [12](#)
- precache, iSEEindexHttpsResource-method
  - (iSEEindexHttpsResource-class), [7](#)
- precache, iSEEindexLocalhostResource-method
  - (iSEEindexLocalhostResource-class), [8](#)
- precache, iSEEindexRcallResource-method
  - (iSEEindexRcallResource-class), [10](#)
- precache, iSEEindexResource-method
  - (iSEEindexResource-class), [11](#)
- precache, iSEEindexRunrResource-method
  - (iSEEindexRunrResource-class), [13](#)
- precache, iSEEindexS3Resource-method
  - (iSEEindexS3Resource-class), [14](#)
- show, iSEEindexResource-method
  - (iSEEindexResource-class), [11](#)