

Package ‘rcoleo’

June 22, 2018

Title rcoleo

Version 0.0.0.9000

Description rcoleo un paquet R permettant le retrait des données collectées dans le cadre du programme de suivi de la biodiversité. Ce paquet R s'appuie sur l'API de Coléo `coléo()`, un système d'information sur la biodiversité du Québec.

Depends R (>= 3.4.2)

License MIT

Encoding UTF-8

LazyData true

Imports RPostgreSQL,
httr,
jsonlite,
stringr

Suggests knitr,
rmarkdown

VignetteBuilder knitr

RoxygenNote 6.0.1

Roxygen list(markdown = TRUE)

R topics documented:

cells	2
get_campaigns	2
get_cells	3
get_gen	4
get_sites	5
get_species	5
get_techs	6
post_campaigns	7
post_cells	7
post_gen	8
post_landmarks	8
post_sites	9

Index**10**

cells*Plant and pollinator example network*

Description

Cells est un jeu de données contenant les 9 000 cellules géoréférencées du plan d'échantillonnage établie en 2016

Usage

```
data(cells)
```

Format

Une liste équivalente au format JSON pour des injections dans l'API de Coléo

Details

- cell_code: Code d'identification unique de la cellule du plan d'échantillonnage (TODO: Ref Samson 2016)
- name: Nom unique de la cellule (si la cellule a fait l'objet de la mise en place de site).
- shp: Fichier de forme de la cellule

Author(s)

MFFP 2017

get_campaigns*Obtenir l'ensemble des campagnes attachées à un site*

Description

Obtenir l'ensemble des campagnes attachées à un site

Usage

```
get_campaigns(site_code = NULL, opened_at = NULL, closed_at = NULL,  
              type = NULL, ...)
```

Arguments

site_code	character vecteur contenant les codes de sites pour lesquelles ont désire récupérer les campagnes (ex. 134_111_F01)
opened_at	character vecteur contenant les dates de début de campagnes (ex. 2017-01-30)
closed_at	character vecteur contenant les dates de fin de campagnes (ex. 2017-01-30)
type	character vecteur contenant le type de campagnes d'inventaires réalisé (ex. végétation)
...	httr options; arguments de la fonction <code>httr::GET()</code>

Examples

```
get_campaigns()
```

`get_cells`

Obtenir les informations sur les cellules depuis l'API de coleo

Description

Obtenir les informations sur les cellules depuis l'API de coleo

Usage

```
get_cells(cell_code = NULL, ...)
```

Arguments

cell_code	character vecteur contenant les identifiants uniques que l'on désire obtenir. Si <code>cell_code</code> n'est pas spécifié, la fonction retournera l'ensemble des entrées présentes dans la table <code>cells</code> .
...	httr options; arguments de la fonction <code>httr::GET()</code>

Value

Retourne un objet de type `list` contenant les réponses de l'API. Chaque niveau de la liste correspond à une page. Pour chacun des appels sur l'API (page), la classe retournée est `getSuccess` ou `getError`. Une réponse de classe `getSuccess` est une liste à deux niveaux composé du contenu (`body`), et la réponse [httr::response](#). Une réponse de classe `getError` dispose de la même structure mais ne contiendra pas de `body`, seulement la réponse de l'API.

Examples

```
get_cells(cell_code=c('111_91'))  
get_cells()
```

get_gen	<i>Fonction générique pour retirer de l'information depuis l'API de Coléo</i>
---------	---

Description

Fonction générique pour retirer de l'information depuis l'API de Coléo

Usage

```
get_gen(endpoint = NULL, query = NULL, ...)
```

Arguments

endpoint	character désignant le point d'entrée pour le retrait des données. Un point d'entrée peut être vu comme une table de la base de données.
query	list de paramètres à passer avec l'appel sur le endpoint.
...	httr options; arguments de la fonction <code>httr::GET()</code>

Details

Les points d'accès de l'API sont énuméré dans l'environnement de coléo, voir `print(rce$endpoints)`

Value

Retourne une objet de type `list` contenant les réponses de l'API. Chaque niveau de la liste correspond à une page. Pour chacun des appels sur l'API (page), la classe retourné est `getSuccess` ou `getError`. Une réponse de classe `getSuccess` est une liste à deux niveaux composé du contenu (body), et la réponse [httr::response](#). Une réponse de classe `getError` dispose de la même structure mais ne contiendra pas de body, seulement la réponse de l'API.

Examples

```
resp <- get_gen(rce$endpoints$cells)
length(resp) # Nombre de pages retourné par l'appel sur le point d'accès de l'API.
str(resp[[1]])
class(resp[[1]])
```

`get_sites`*Obtenir les informations sur les sites depuis l'API de coleo*

Description

Obtenir les informations sur les sites depuis l'API de coleo

Usage

```
get_sites(site_code = NULL, ...)
```

Arguments

<code>site_code</code>	character vecteur contenant les identifiants uniques que l'on désire obtenir. Si <code>site_code</code> n'est pas spécifié, la fonction retournera l'ensemble des entrées présentes dans la table <code>cells</code> .
<code>...</code>	httr options; arguments de la fonction <code>httr::GET()</code>

Value

Retourne un objet de type `list` contenant les réponses de l'API. Chaque niveau de la liste correspond à une page. Pour chacun des appels sur l'API (page), la classe retournée est `getSuccess` ou `getError`. Une réponse de classe `getSuccess` est une liste à deux niveaux composé du contenu (`body`), et la réponse `httr::response`. Une réponse de classe `getError` dispose de la même structure mais ne contiendra pas de `body`, seulement la réponse de l'API.

Examples

```
get_sites(site_code=c('135_104_H01'))  
get_sites()
```

`get_species`*Retourne les entrées d'espèces présentes dans la base de données*

Description

Retourne les entrées d'espèces présentes dans la base de données

Usage

```
get_species(searchEntry = NULL, species = NULL, genus = NULL,  
            vernacular = NULL, ...)
```

Arguments

searchEntry	character vecteur contenant les mots clés (recherche sur l'intégralité de la table)
species	character vecteur contenant les espèces recherché (sensible à la case) - Obsolète si l'argument query est spécifié
genus	character vecteur contenant les genres d'espèce recherché (sensible à la case) - Obsolète si l'argument query est spécifié
vernacular	character vecteur contenant les noms vernaculaires des espèces recherchées (sensible à la case) - Obsolète si l'argument query est spécifié
...	httr options; arguments de la fonction <code>httr::GET()</code>

Value

Retourne un objet de type `list` contenant les réponses de l'API. Chaque niveau de la liste correspond à une page. Pour chacun des appels sur l'API (page), la classe retournée est `getSuccess` ou `getError`. Une réponse de classe `getSuccess` est une liste à deux niveaux composée du contenu (body), et la réponse [httr::response](#). Une réponse de classe `getError` dispose de la même structure mais ne contiendra pas de body, seulement la réponse de l'API.

Examples

```
get_species(vernacular=c('Érable'))
get_species()
```

get_techs

Obtenir l'ensemble des techniciens attachés à une campagne

Description

Obtenir l'ensemble des techniciens attachés à une campagne

Usage

```
get_techs(name = NULL, lastname = NULL, ...)
```

Arguments

name	character vecteur contenant le prénom du technicien (sensible à la case)
lastname	character vecteur contenant le nom du technicien (sensible à la case)
...	httr options; arguments de la fonction <code>httr::GET()</code>

Value

Retourne un objet de type `list` contenant les réponses de l'API. Chaque niveau de la liste correspond à une page. Pour chacun des appels sur l'API (page), la classe retournée est `getSuccess` ou `getError`. Une réponse de classe `getSuccess` est une liste à deux niveaux composée du contenu (body), et la réponse [httr::response](#). Une réponse de classe `getError` dispose de la même structure mais ne contiendra pas de body, seulement la réponse de l'API.

Examples

```
get_techs(name="Caroline")
```

`post_campaigns`*Publication d'une campagne sur la base de données de Coléo*

Description

Cette fonction applique la méthode POST sur le point d'entrées campaigns de l'API de Coleo

Usage

```
post_campaigns(data, ...)
```

Arguments

<code>data</code>	une liste ou chacun des niveaux correspond aux données attribuées au ednpoint.
<code>...</code>	httr options; arguments de la fonction <code>httr::POST()</code>

`post_cells`*Publication d'une cellule sur la base de données de Coléo*

Description

Cette fonction applique la méthode POST sur le point d'entrées cells de l'API de Coleo

Usage

```
post_cells(data, ...)
```

Arguments

<code>data</code>	une liste ou chacun des niveaux correspond aux données attribuées au ednpoint.
<code>...</code>	httr options; arguments de la fonction <code>httr::POST()</code>

Value

Un objet `list`, dont chacun des niveaux correspond à la réponse de l'API. La réponse peut être de classe `postError` ou `postSuccess`.

See Also

[post_gen](#) pour la structure de sortie de la fonction.

post_gen	<i>Fonction générique pour envoyer de l'information vers l'API de Coléo</i>
----------	---

Description

Fonction générique pour envoyer de l'information vers l'API de Coléo

Usage

```
post_gen(endpoint, singleton, ...)
```

Arguments

endpoint	Point d'entrée pour le retrait des données.
singleton	Données à injecter à l'intérieur de la base de données.
...	httr options; arguments de la fonction <code>httr::POST()</code>

Value

Retourne un objet de type `list` contenant les réponses de l'API. Chaque niveau de la liste correspond à la réponse de l'API. Pour chacun des appels POST sur l'API, la classe retournée est `postSuccess` ou `postError`. Une réponse de classe `postSuccess` ou `postError` est une liste à deux niveaux composée du `body`, et de la réponse [httr::response](#). Si la réponse est de classe `postSuccess`, le `body` contiendra les données injectées sinon il contiendra le message d'erreur retourné par l'API.

post_landmarks	<i>Publication d'un landmark dans l'API de coléo</i>
----------------	--

Description

Cette fonction applique la méthode POST sur le point d'entrées sites de l'API de Coleo

Usage

```
post_landmarks(data, ...)
```

Arguments

data	une liste où chacun des niveaux correspond aux données attribuées au endpoint.
...	httr options; arguments de la fonction <code>httr::POST()</code>

post_sites*Publication d'un site sur la base de données de Coléo*

Description

Cette fonction applique la méthode POST sur le point d'entrées sites de l'API de Coleo

Usage

```
post_sites(data, ...)
```

Arguments

data	une liste ou chacun des niveaux correspond aux données attribuées au ednpoint.
...	httr options; arguments de la fonction <code>httr::POST()</code>

Index

*Topic **datasets**

cells, [2](#)

cells, [2](#)

get_campaigns, [2](#)

get_cells, [3](#)

get_gen, [4](#)

get_sites, [5](#)

get_species, [5](#)

get_techs, [6](#)

httr::response, [3–6, 8](#)

post_campaigns, [7](#)

post_cells, [7](#)

post_gen, [7, 8](#)

post_landmarks, [8](#)

post_sites, [9](#)