

# Package ‘waruDB’

September 18, 2019

**Type** Package

**Title** Database backend functions for waruPortal

**Version** 0.1.0

**Description** waruDB contains all the functions that are used within waruPortal to access data stored in the MySQL database backend.

**Depends** magrittr

**Imports** purrr, pool, dplyr, openssl, stringr, lubridate, uuid, tibble

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Author** Tom Wilson [aut, cre]

**Maintainer** Tom Wilson <tpw2@aber.ac.uk>

## R topics documented:

bioBankHash . . . . .	2
checkSampleTrackerHash . . . . .	3
contactMandatory . . . . .	3
DBSform . . . . .	4
deleteSampleTracker . . . . .	4
getDashboardName . . . . .	5
getRegistrationHashCodes . . . . .	5
getSourceComponents . . . . .	6
loadBioBank . . . . .	6
loadBioBankIndex . . . . .	7
loadParticipantRegistration . . . . .	7
loadProjectIndex . . . . .	8
loadRecruitmentContacts . . . . .	8
loadSampleTracker . . . . .	9
loadSampleTrackerAll . . . . .	9

loadSampleTrackerDelete . . . . .	10
loadUserData . . . . .	10
prepareBioBank . . . . .	11
prepareContactForm . . . . .	11
prepareContactFormBatch . . . . .	12
prepareRegistrationForm . . . . .	12
prepareSampleTracker . . . . .	13
registrationHash . . . . .	13
registrationMandatory . . . . .	14
sampleTrackerHash . . . . .	14
sampleTrackerMandatory . . . . .	15
saveBioBank . . . . .	15
saveContactForm . . . . .	16
saveRegistrationForm . . . . .	16
saveSampleTracker . . . . .	17
saveSampleTrackerDelete . . . . .	17
validateBioBank . . . . .	18
validateContactForm . . . . .	18
validateSampleTracker . . . . .	19
<b>Index</b>	<b>20</b>

---

bioBankHash	<i>BioBank Hash Code</i>
-------------	--------------------------

---

**Description**

A simple generation of a hash code which is used to check if a new BioBank record already has an existing entry in the database.

**Usage**

```
bioBankHash(inFile)
```

**Arguments**

inFile                    a ‘tibble’ from the file input of BioBank batch upload

**Details**

The hash code is simply the base64 encoded concatantion of; ‘Sample ID’, ‘BioFluid’ and ‘SampleContainer’

**Value**

a character vector of the base64 encoded BioBank hash codes

---

`checkSampleTrackerHash`*Check Sample Tracker Hash*

---

**Description**

Check if the incoming Sample Tracker entry already exists in the database

**Usage**

```
checkSampleTrackerHash(formData, db_connection)
```

**Arguments**

`formData`            a character of the reactive output from the Sample Tracker form entry  
`db_connection`    a 'dbPool' or 'RMariaDB' database connection

**Value**

a numeric value either 0 (hash already exists) 1 (incoming hash is new)

---

`contactMandatory`*Mandatory Value Check (Contact Form)*

---

**Description**

Mandatory Value Check (Contact Form)

**Usage**

```
contactMandatory(inputData)
```

**Arguments**

`inputData`            list of input data from Participant Interest contact form

**Value**

a numeric value either 0;missing mandatory values or 1; all values present

---

DBSform	<i>DBS Form Data</i>
---------	----------------------

---

**Description**

Extract Sample Tracker data for Dried Blood Spot (DBS) cards

**Usage**

```
DBSform(formData)
```

**Arguments**

formData	a character of the reactive output from the Sample Tracker form entry
----------	---

**Value**

return a character of 'formData' for DBS cards

---

deleteSampleTracker	<i>Delete Sample Tracker</i>
---------------------	------------------------------

---

**Description**

Delete Sample Tracker

**Usage**

```
deleteSampleTracker(db_connection, hash)
```

**Arguments**

db_connection	a 'dbPool' or 'RMariaDB' database connection
hash	a character string of the 'hash' for the entry to delete

---

getDashboardName	<i>Get Dashboard Name</i>
------------------	---------------------------

---

**Description**

Retrieve the name of the correct login dashbaord to source based the User's access level

**Usage**

```
getDashboardName(db_connection, username)
```

**Arguments**

db_connection	a 'dbPool' or 'RMariaDB' database connection
username	a character string of a valid 'username'

**Value**

a character string of the dashbaord to source

---

getRegistrationHashCodes
<i>Get Registration Hash Codes</i>

---

**Description**

Get Registration Hash Codes

**Usage**

```
getRegistrationHashCodes(db_connection)
```

**Arguments**

db_connection	a 'dbPool' or 'RMariaDB' database connection
---------------	--

**Value**

a character vector of hash codes

---

getSourceComponents	<i>Get Source Components</i>
---------------------	------------------------------

---

**Description**

Retrieve the components to source at login based on the User's access level

**Usage**

```
getSourceComponents(db_connection, username)
```

**Arguments**

db_connection	a 'dbPool' or 'RMariaDB' database connection
username	a character string of a valid 'username'

**Value**

a vector of components to source at login

---

loadBioBank	<i>Load BioBank</i>
-------------	---------------------

---

**Description**

Load the BioBak table

**Usage**

```
loadBioBank(db_connection)
```

**Arguments**

db_connection	a 'dbPool' or 'RMariaDB' database connection
---------------	--

**Value**

a 'tibble' of the complete BioBank table

---

loadBioBankIndex	<i>Load BioBank Index</i>
------------------	---------------------------

---

**Description**

Load the BioBank Index table

**Usage**

```
loadBioBankIndex(db_connection)
```

**Arguments**

db\_connection    a 'dbPool' or 'RMariaDB' database connection

**Value**

a 'tibble' of BioBank box descriptions

---

loadParticipantRegistration	<i>Load Participant Registration Data</i>
-----------------------------	---

---

**Description**

Load Participant Registration Data

**Usage**

```
loadParticipantRegistration(db_connection)
```

**Arguments**

db\_connection    a 'dbPool' or 'RMariaDB' database connection

**Value**

a 'tibble' of registered participants

---

loadProjectIndex	<i>Load Project Index</i>
------------------	---------------------------

---

**Description**

Load the Project Index table

**Usage**

```
loadProjectIndex(db_connection)
```

**Arguments**

db\_connection    a 'dbPool' or 'RMariaDB' database connection

**Value**

a 'tibble' of WARU projects

---

loadRecruitmentContacts	<i>Load Recruitment Contacts</i>
-------------------------	----------------------------------

---

**Description**

Load all data from the 'Participant Recruitment Database'

**Usage**

```
loadRecruitmentContacts(db_connection)
```

**Arguments**

db\_connection    a 'dbPool' or 'RMariaDB' database connection

**Value**

a 'tibble' of the 'Participant Recruitment Database'



---

loadSampleTracker	<i>Load Sample Tracker Data</i>
-------------------	---------------------------------

---

**Description**

Load useful data from \*SampleTracker\* table

**Usage**

```
loadSampleTracker(db_connection)
```

**Arguments**

db\_connection    a 'dbPool' or 'RMariaDB' database connection

**Value**

a 'tibble' of sample details

---

loadSampleTrackerAll	<i>Load All Sample Tracker</i>
----------------------	--------------------------------

---

**Description**

Load All Sample Tracker

**Usage**

```
loadSampleTrackerAll(db_connection)
```

**Arguments**

db\_connection    a 'dbPool' or 'RMariaDB' database connection

**Value**

a 'tibble' of the entire 'SampleTracker' table

---

`loadSampleTrackerDelete`*Load Sample Tracker (Deleted) Data*

---

**Description**

Load useful data from \*SampleTrackerDelete\* table

**Usage**

```
loadSampleTrackerDelete(db_connection)
```

**Arguments**

`db_connection` a 'dbPool' or 'RMariaDB' database connection

**Value**

a 'tibble' of deleted sample details

---

`loadUserData`*Load User Data*

---

**Description**

Load all active users from \*Users\* table. Active refers to users who are currently registered for use on the system; not user who are currently active within a session.

**Usage**

```
loadUserData(db_connection)
```

**Arguments**

`db_connection` a 'dbPool' or 'RMariaDB' database connection

**Value**

a 'tibble' of active users

---

prepareBioBank	<i>Prepare BioBank</i>
----------------	------------------------

---

**Description**

Preparing incoming BioBank batch file for upload to the database

**Usage**

```
prepareBioBank(inFile, username)
```

**Arguments**

inFile	a 'tibble' from the file input
username	a character of the current session user (ie, 'USER\$name')

**Value**

a 'tibble' of the prepared 'inFile'

---

prepareContactForm	<i>Prepare Contact Information Form</i>
--------------------	---

---

**Description**

Prepare incoming Contact Form data for input into the database

**Usage**

```
prepareContactForm(formData, username)
```

**Arguments**

formData	a character of the reactive output from the Contact Form entry
username	a character of the current session user (ie, 'USER\$name')

**Value**

a list of prepared form data

---

`prepareContactFormBatch`*Prepare Contact Information Form (Batch Mode)*

---

**Description**

Prepare incoming Contact Form data for batch input into the database

**Usage**

```
prepareContactFormBatch(inFile, username)
```

**Arguments**

<code>inFile</code>	a 'tibble' from the file input
<code>username</code>	a character of the current session user (ie, 'USER\$name')

**Value**

a list of prepared form data

---

`prepareRegistrationForm`*Prepare New Participant Registration Form*

---

**Description**

Prepare incoming New Participant Registration Form data for input into the database

**Usage**

```
prepareRegistrationForm(formData, username)
```

**Arguments**

<code>formData</code>	a character of the reactive output from the Registration Form entry
<code>username</code>	a character of the current session user (ie, 'USER\$name')

**Value**

a list of prepared form data

---

prepareSampleTracker	<i>Prepare Sample Tracker Form</i>
----------------------	------------------------------------

---

**Description**

Prepare incoming Sample Tracker form data for input into the database

**Usage**

```
prepareSampleTracker(formData, username)
```

**Arguments**

formData	a character of the reactive output from the Sample Tracker form entry
username	a character of the current session user (ie, 'USER\$name')

**Value**

a one or two element list of prepared 'formData'. One list element for each sample type (Urine & DBS)

---

registrationHash	<i>Registration Form Hash Code</i>
------------------	------------------------------------

---

**Description**

A simple generation of a hash code which is used to check if newly registered participant has an existing entry in the database.

**Usage**

```
registrationHash(formData)
```

**Arguments**

formData	a character of the reactive output from the Registration Form entry
----------	---

**Value**

a character string of the base64 encoded hash

---

registrationMandatory    *Mandatory Value Check (Participant Registration Form)*

---

**Description**

Mandatory Value Check (Participant Registration Form)

**Usage**

```
registrationMandatory(inputData)
```

**Arguments**

inputData            list of input data from Participant Registration form

**Value**

a numeric value either 0;missing mandatory values or 1; all values present

---

sampleTrackerHash        *Sample Tracker Hash Code*

---

**Description**

A simple generation of a hash code which is used to check if an incoming sample already has an existing entry in the database.

**Usage**

```
sampleTrackerHash(formData)
```

**Arguments**

formData            a character of the reactive output from the Sample Tracker form entry

**Details**

The hash code is simply the base64 encoded concatenation of; 'Sample ID', 'Date Received' and 'Sample Type'

**Value**

a character of the base64 encoded sample hash code

---

`sampleTrackerMandatory`*Mandatory Value Check (Sample Tracker)*

---

**Description**

Mandatory Value Check (Sample Tracker)

**Usage**

```
sampleTrackerMandatory(inputData)
```

**Arguments**

`inputData`      list of input data from Sample Tracker form

**Value**

a numeric value either 0;missing mandatory values or 1; all values present

---

`saveBioBank`*Save BioBank Data*

---

**Description**

Save BioBank records to the database

**Usage**

```
saveBioBank(db_connection, inFile)
```

**Arguments**

`db_connection`    a 'dbPool' or 'RMariaDB' database connection

`inFile`            the 'tibble' output from 'prepareBioBank'

---

saveContactForm	<i>Save Contact Form</i>
-----------------	--------------------------

---

**Description**

Save data captured from a form entry reactive into the \*ParticipantInterest\* table

**Usage**

```
saveContactForm(formData, db_connection)
```

**Arguments**

formData	the reactive output from 'formData'
db_connection	a 'dbPool' or 'RMariaDB' database connection

---

saveRegistrationForm	<i>Register New Participant</i>
----------------------	---------------------------------

---

**Description**

Register New Participant

**Usage**

```
saveRegistrationForm(formData, db_connection)
```

**Arguments**

formData	the 'list' output from 'prepareRegistrationForm'
db_connection	a 'dbPool' or 'RMariaDB' database connection

**Value**

a character string of the allocated ParticipantID



---

saveSampleTracker	<i>Save Sample Tracker</i>
-------------------	----------------------------

---

**Description**

Save data captured from a form entry reactive into the \*SampleTracker\* table

**Usage**

```
saveSampleTracker(formData, db_connection)
```

**Arguments**

formData	the reactive output from 'formData'
db_connection	a 'dbPool' or 'RMariaDB' database connection

---

saveSampleTrackerDelete	<i>Save Sample Tracker (Delete)</i>
-------------------------	-------------------------------------

---

**Description**

Save Sample Tracker (Delete)

**Usage**

```
saveSampleTrackerDelete(db_connection, inFile, username)
```

**Arguments**

db_connection	a 'dbPool' or 'RMariaDB' database connection
inFile	the 'tibble' output from 'loadSampleTracker'
username	a character of the current session user (ie, 'USER\$name')

---

validateBioBank	<i>Validate Sample Tracker (Batch Upload)</i>
-----------------	---

---

**Description**

Validate the BioBank batch upload file

**Usage**

```
validateBioBank(inFile, db_connection)
```

**Arguments**

inFile	a 'tibble' from the file input
db_connection	a 'dbPool' or 'RMariaDB' database connection

**Value**

a numeric value either 0;validation failed or 1; validation successful

---

validateContactForm	<i>Validate Contact Form (Batch Upload)</i>
---------------------	---

---

**Description**

Validator function for when batch upload is used instead of standard form entry

**Usage**

```
validateContactForm(inFile)
```

**Arguments**

inFile	a 'tibble' from the file input
--------	--------------------------------

**Value**

a numeric value either 0;validation failed or 1; validation successful

---

`validateSampleTracker` *Validate Sample Tracker (Batch Upload)*

---

**Description**

Validator function for when batch upload is used instead of standard form entry

**Usage**

```
validateSampleTracker(inFile, db_connection)
```

**Arguments**

`inFile`            a 'tibble' from the file input  
`db_connection`    a 'dbPool' or 'RMariaDB' database connection

**Value**

a numeric value either 0;validation failed or 1; validation successful

# Index

bioBankHash, [2](#)

checkSampleTrackerHash, [3](#)  
contactMandatory, [3](#)

DBSform, [4](#)  
deleteSampleTracker, [4](#)

getDashboardName, [5](#)  
getRegistrationHashCodes, [5](#)  
getSourceComponents, [6](#)

loadBioBank, [6](#)  
loadBioBankIndex, [7](#)  
loadParticipantRegistration, [7](#)  
loadProjectIndex, [8](#)  
loadRecruitmentContacts, [8](#)  
loadSampleTracker, [9](#)  
loadSampleTrackerAll, [9](#)  
loadSampleTrackerDelete, [10](#)  
loadUserData, [10](#)

prepareBioBank, [11](#)  
prepareContactForm, [11](#)  
prepareContactFormBatch, [12](#)  
prepareRegistrationForm, [12](#)  
prepareSampleTracker, [13](#)

registrationHash, [13](#)  
registrationMandatory, [14](#)

sampleTrackerHash, [14](#)  
sampleTrackerMandatory, [15](#)  
saveBioBank, [15](#)  
saveContactForm, [16](#)  
saveRegistrationForm, [16](#)  
saveSampleTracker, [17](#)  
saveSampleTrackerDelete, [17](#)

validateBioBank, [18](#)  
validateContactForm, [18](#)  
validateSampleTracker, [19](#)