

# Package: PhysioEcosystem (via r-universe)

October 3, 2026

**Title** Unified Entry Point for the Physio Ecosystem (Meta-Package)

**Version** 1.0.1

**Description** A meta-package for the Physio ecosystem: attaching it loads and re-exports the public API of the foundation and the core analysis packages (PhysioExperiment shared data model, PhysioIO file/database I/O, PhysioPreprocess signal preprocessing, and PhysioAnalysis time-frequency/connectivity/network/statistics/visualization), so existing code that used the former monolithic 'PhysioExperiment' meta-package keeps working. It also provides the GUI/REST launchers. Domain packages (EEG, ECG, EMG, EDA, MoCap, cross-modal, MSK-Net, OpenSim, clinical) are installed separately from the same ecosystem.

**Depends** R (>= 4.2), PhysioExperiment (>= 2.0.0), PhysioIO, PhysioPreprocess, PhysioAnalysis

**Imports** methods, utils

**Suggests** jsonlite, testthat (>= 3.2.0), knitr, rmarkdown, pkgdown, plumber (>= 1.2.0), later, callr, covr

**biocViews** Software, TimeCourse, Preprocessing

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.3

**VignetteBuilder** knitr

**URL** <https://github.com/x-biosignal/PhysioEcosystem>,  
<https://x-biosignal.r-universe.dev/PhysioEcosystem>

**BugReports** <https://github.com/x-biosignal/PhysioEcosystem/issues>

**Collate** 'PhysioEcosystem-package.R' 'reexports.R' 'gui-launcher.R'  
'zzz.R'

**Config/pak/sysreqs** libssl-dev zlib1g-dev

**Repository** https://x-biosignal.r-universe.dev  
**Date/Publication** 2026-10-03 13:11:41 UTC  
**RemoteUrl** https://github.com/x-biosignal/PhysioEcosystem  
**RemoteRef** HEAD  
**RemoteSha** 3a6a8e7ebe4ec47a97e3ad2be9ba0aa63086c940

Contents

|                                |          |
|--------------------------------|----------|
| checkGUIDependencies . . . . . | 2        |
| launchGUI . . . . .            | 3        |
| startAPIServer . . . . .       | 4        |
| <b>Index</b>                   | <b>6</b> |

---

|                      |                               |
|----------------------|-------------------------------|
| checkGUIDependencies | <i>Check GUI Dependencies</i> |
|----------------------|-------------------------------|

---

Description

Checks if all required dependencies for the GUI are available.

Usage

checkGUIDependencies()

Value

A list with the status of each dependency.

Examples

```
# Report which GUI/REST back ends are available in this session:  
checkGUIDependencies()
```

---

`launchGUI`*Launch PhysioExperiment GUI*

---

## Description

Starts the PhysioExperiment graphical user interface. This launches a local web server with the Plumber API backend and opens the React-based GUI in either a browser or Electron desktop application.

## Usage

```
launchGUI(  
  port = 8000L,  
  host = "127.0.0.1",  
  desktop = FALSE,  
  browser = TRUE,  
  quiet = FALSE  
)
```

## Arguments

|                      |                                                                                                                              |
|----------------------|------------------------------------------------------------------------------------------------------------------------------|
| <code>port</code>    | Integer. Port number for the API server. Default is 8000.                                                                    |
| <code>host</code>    | Character. Host address to bind to. Default is "127.0.0.1" for local access only. Use "0.0.0.0" to allow external access.    |
| <code>desktop</code> | Logical. If TRUE, attempts to launch the Electron desktop application. If FALSE (default), opens in the default web browser. |
| <code>browser</code> | Logical. If TRUE (default), automatically opens the GUI in the browser. Set to FALSE to start only the API server.           |
| <code>quiet</code>   | Logical. If TRUE, suppresses startup messages. Default is FALSE.                                                             |

## Details

The GUI provides a modern, responsive interface for:

- Data import and management (EDF, BDF, BrainVision, HDF5, CSV)
- Signal visualization with multi-channel display
- Preprocessing (filtering, referencing, resampling)
- Time-frequency analysis (spectrograms, wavelets)
- Connectivity analysis (coherence, PLV, PLI)
- Statistical testing (t-test, ANOVA, cluster permutation)
- Workflow building for batch processing

The GUI communicates with R through a REST API built on plumber. All computations are performed in R using the PhysioExperiment package.

**Value**

Invisibly returns the plumber object. The function blocks while the server is running unless run in background mode.

**Desktop Mode**

Desktop mode requires Electron to be installed. The Electron application provides native file access and a more integrated desktop experience. If Electron is not available, the function falls back to browser mode.

**Requirements**

The following packages are required and will be loaded automatically:

- plumber - for the REST API server
- jsonlite - for JSON serialization

**Examples**

```
# Check that the GUI/REST prerequisites are installed before launching:
checkGUIDependencies()

## Not run:
# Launch GUI in browser
launchGUI()

# Launch on a different port
launchGUI(port = 3000)

# Start API server without opening browser
launchGUI(browser = FALSE)

# Launch in desktop mode (requires Electron)
launchGUI(desktop = TRUE)

## End(Not run)
```

---

startAPIServer

---

*Start PhysioExperiment API Server (Non-blocking)*


---

**Description**

Starts the API server in the background, allowing R to remain interactive. This is useful for development and testing.

**Usage**

```
startAPIServer(port = 8000L, host = "127.0.0.1", quiet = FALSE)
```

**Arguments**

|                    |                                          |
|--------------------|------------------------------------------|
| <code>port</code>  | Integer. Port number for the API server. |
| <code>host</code>  | Character. Host address to bind to.      |
| <code>quiet</code> | Logical. Suppress startup messages.      |

**Value**

A plumber background process handle that can be used to stop the server with `$kill()`.

**Examples**

```
# The background REST server needs the plumber and callr packages:
checkGUIDependencies()

## Not run:
# Start server in background
server <- startAPIServer()

# Do other work...

# Stop server
server$kill()

## End(Not run)
```

# Index

checkGUIDependencies, [2](#)

launchGUI, [3](#)

startAPIServer, [4](#)