

Proposal for mealR: An R Package for TheMealDB API

1. Introduction and Motivation

In this proposal, we plan to use TheMealDB database to get access to a large number of meal recipes, ingredients and cooking instructions via a RESTful API. Although it is accessible from standard HTTP requests, there is no dedicated R package that incorporates the database's functionality into the R environment in the current stage.

The mealR package we plan to develop aims to address this problem by providing a structured and user-friendly API wrapper that will make users retrieve the meal recipes much easier. This package will be especially useful for interested homemakers and food researchers working on alimentary projects, such as nutrition analysis, meal recommendation systems, and interactive Shiny applications.

The key motivations for developing mealR include:

- Simplifying access to TheMealDB API within R.
- Providing structured, well-documented functions for fetching meal data.
- Ensuring robust error handling and stable API calls.
- Returning results in data.frame format to facilitate data manipulation and visualization.

2. Objectives

The main goals of the mealR package are:

1. To develop a fully functional R package that wraps TheMealDB API.
2. To implement a set of well-structured functions for accessing meals, categories, ingredients, and regions.
3. To ensure error handling mechanisms for API failures and unexpected responses.
4. To follow best practices in R package development, including documentation and unit testing.
5. To provide a package that is installable via GitHub and, if feasible, CRAN.

3. Features and Functionality

The package will include the following core functions:

Function Name	Description
search_meal(query)	Search for a meal by name.
get_random_meal()	Retrieve a randomly selected meal.
get_meal_by_id(id)	Fetch detailed meal information using a unique ID.

get_categories()	List all available meal categories.
get_meals_by_category(category)	Retrieve meals belonging to a specific category.
get_meals_by_region(region)	Retrieve meals based on a specified region or country.

4. Methodology

API Integration

- The package will use established R libraries for API requests and JSON parsing, ensuring seamless data retrieval and conversion.
- The package will implement error handling using tryCatch() to manage connection failures and invalid API responses.

Package Development

- The package structure will be created using devtools::create_package("mealR").
- Function definitions will be organized in separate script files within the R/ directory.
- roxygen2 will be used to generate documentation and NAMESPACE files.

Testing Strategy

- Unit tests will be implemented using testthat to verify function correctness and API response stability.

Continuous Integration

- GitHub Actions will be configured to automate testing across different operating systems.
- The CI workflow will check for package validity, test coverage, and compliance with CRAN standards.

5. Development Plan

Phase	Tasks	Time (days)	line
Setup & Development	Create repository, define API request functions, implement core functions, and test.	5	
Documentation & Testing	Write documentation, implement unit tests.	5	

Deployment & Feedback	Publish on GitHub, gather feedback, refine the package, and prepare for CRAN submission.	4
-----------------------	--	---

6. Challenges and Solutions

Challenge	Proposed Solution
API rate limits	Implement caching using memoise to reduce redundant API calls.
Handling API failures	Use tryCatch() for error handling and meaningful error messages.

7. Conclusion

The mealR package's goal is to provide an easy-to-implement and use interface for TheMealDB API, which is going to make processes such as data retrieval and analysis of meal recipes more seamlessly. We plan to follow the practices in R package development, including structured documentation, unit testing, and CI/CD, this project will contribute a valuable tool to the R community.

The package's functionality is not limited only to meal data retrieval simplification, it will also support integration into Shiny dashboards, machine learning models and data visualisation projects. By making mealR available via GitHub and potentially CRAN, we hope its functionality can be tested and utilized by a range of users in both research and homemaking applications.

References

- TheMealDB API Documentation: <https://www.themealdb.com/api.php>
- CRAN Package Development Guide: <https://cran.r-project.org/doc/manuals/r-release/R-exts.html>
- GitHub Actions for R: <https://github.com/r-lib/actions>