

R API Integration Project

Overview

In this two-session project, you'll explore real-world APIs by connecting to one, retrieving data, and analyzing it using R, tidyverse, and SQL. **Follow your interests and the interests of your team** - build something you actually want to use!

While you're encouraged to collaborate with teammates, **each student must submit their own individual work at the end of each session.**

AI tools (ChatGPT, Claude, GitHub Copilot, etc.) are strongly encouraged! Use them to help you understand concepts, debug code, and learn faster.

Project Philosophy

Build something useful and meaningful. Create an interface or tool that you (or others) might actually want to use in the future. Think about:

- What data interests you?
- What questions do you want answered?
- What would make your life easier or more informed?
- What would be fun or engaging to explore?

This is **your** project - make it reflect your interests and creativity!

Core Requirements

Your project must incorporate:

1. **API Integration** - Connect to a real-world API and retrieve data
2. **Tidyverse (dplyr/tidyr)** - Process and transform your data
3. **SQL/RSQLite** - Store and query your data
4. **Security** - Handle API keys appropriately (never hardcode them!)

Beyond these requirements, the implementation is up to you.

Getting Started

Choose Your Data Source

APIs that don't require keys:

- REST Countries - <https://restcountries.com/>
- PokéAPI - <https://pokeapi.co/>
- Open Library - <https://openlibrary.org/developers/api>

APIs with free registration:

- OpenWeatherMap - <https://openweathermap.org/api>
- News API - <https://newsapi.org/>
- NASA APIs - <https://api.nasa.gov/>
- CoinGecko - <https://www.coingecko.com/en/api>
- Alpha Vantage - <https://www.alphavantage.co/>
- TMDb - <https://www.themoviedb.org/>
- Spotify - <https://developer.spotify.com/>

Or find your own! There are thousands of free APIs - pick something that interests you.

Think About Usefulness

What would make this project valuable to you or others?

- Does it answer questions you care about?
- Could you run it again with new data?
- Does it present information in a useful way?
- Does it solve a problem (even a small one)?

Timeline

Session 1 (Due at end of class)

Show meaningful progress. Can you successfully get data from an API?

Session 2 (Due at end of class)

Submit your completed project with full pipeline and documentation.

Suggested Exploration Areas

API Integration

- How do you make HTTP requests in R?
- How do you handle JSON responses?
- What about errors or rate limits?
- Where should API keys be stored?

Data Processing

- How can tidyverse help transform your data?
- What cleaning or filtering might you need?
- How do you wrangle data for analysis?

Database Design

- What tables do you need?
- What fields make sense?
- How do you get R data frames into SQL?
- What queries would reveal interesting insights?

Interface & Usability

- How will someone interact with your project?
- Could you make it easier to use or understand?
- What output format makes sense?

Tools & Libraries

You'll likely need some combination of:

- `httr` or `httr2` - for API calls
- `jsonlite` - for parsing JSON
- `tidyverse` (`dplyr`, `tidyr`, etc.) - for data work
- `RSQLite` and `DBI` - for database operations
- `dotenv` - for managing secrets

How you use them is up to you!

Questions to Guide Your Work

Planning:

- What story do you want your data to tell?
- What insights are you looking for?
- Who might benefit from this tool?

Implementation:

- How can you break this into manageable steps?
- What should you build first?
- When should you test?

Analysis:

- What patterns or trends appear in your data?
- What queries reveal the most interesting information?
- How can you present findings clearly?

Using AI Effectively

AI tools can help you:

- Understand unfamiliar concepts or syntax
- Debug errors and issues
- Learn new techniques
- Explore different approaches
- Write better documentation

But remember: you need to understand what the code does!

Collaboration

Work together on:

- Brainstorming ideas
- Understanding documentation
- Solving problems
- Learning new concepts

Work individually on:

- Writing your own code
- Creating your own implementation
- Submitting your own unique project

Team members can work with the same API, but each submission should be original work.

What to Submit

Include whatever you think best demonstrates your work:

- Your R script(s) (with comments!)
- Documentation explaining your project
- Instructions for running it
- Information about your findings
- Anything else that helps tell your story

Think about: if someone found this project a year from now, could they understand and use it?

Evaluation

Your work will be assessed on:

Session 1: Are you making progress? Do you have a working connection to an API?

Session 2:

- Does your project work end-to-end?
- Did you effectively use tidyverse, SQL, and APIs?
- Is your code secure and well-documented?
- Is there evidence of creative thinking?
- Could someone actually use this?

Some Ideas to Spark Creativity

Personal Tools:

- Something to track prices or trends you care about
- A way to aggregate information from sources you follow
- A tool to help make decisions or recommendations

Learning Projects:

- Explore data about a hobby or interest
- Compare and contrast different options
- Visualize patterns or relationships

Just for Fun:

- Build something quirky or unexpected
- Combine data in unusual ways
- Create something that makes you smile

The best projects often come from genuine curiosity!

Tips

- Start simpler than you think you need to
- Test early and often
- Read documentation (it's usually helpful!)
- Use AI when you're stuck
- Ask questions
- Don't be afraid to experiment
- Focus on working code before perfection

- Your project doesn't need to be complex to be good

Common Challenges

You might encounter:

- API authentication issues
- Rate limiting
- JSON parsing complexities
- Data cleaning needs
- SQL syntax questions
- Error handling
- Documentation decisions

These are all normal! Use your resources (teammates, AI, documentation, instructor) to work through them.

Resources

- Your chosen API's documentation
- R documentation and tidyverse guides
- RSQLite and DBI documentation
- AI assistants
- Stack Overflow
- Your teammates and instructor

Remember: This is an open-ended exploration. There's no single "right" way to do this. The best project is one that interests you, works well, and demonstrates what you've learned. Be creative, be curious, and build something you're proud of!

Good luck - have fun with this!

R-Specific Getting Started Tips

Installing Packages

`r`

```
install.packages(c("httr", "jsonlite", "tidyverse", "RSQLite", "DBI", "dotenv"))
```

Basic API Call Structure

```
r
library(httr)
library(jsonlite)

# Make a GET request
response <- GET("https://api.example.com/data")

# Parse JSON response
data <- fromJSON(content(response, "text"))
...
```

Storing API Keys

Create a `.Renviron` file in your project directory:

...

```
API_KEY=your_key_here
```

Then in your R script:

```
r
api_key <- Sys.getenv("API_KEY")
```

Working with Data

```
r
library(tidyverse)

# Transform your data
df <- data %>%
  filter(condition) %>%
  mutate(new_column = calculation) %>%
  select(relevant_columns)
```

Database Operations

```
r
library(DBI)
library(RSQLite)
```

```
# Connect to database
```

```
con <- dbConnect(RSQLite::SQLite(), "my_database.db")
```

```
# Write data
```

```
dbWriteTable(con, "my_table", df, overwrite = TRUE)
```

```
# Query data
```

```
results <- dbGetQuery(con, "SELECT * FROM my_table WHERE condition")
```

```
# Close connection
```

```
dbDisconnect(con)
```