**REI Full Catalog Scraping Scalability Plan**

**Overview**

This document outlines a systematic approach to capture REI's complete product catalog, ensuring comprehensive SKU coverage while maintaining efficiency and avoiding duplicate processing.

**Phase 1: Category Discovery & Mapping**

**Objective:** Build a complete category hierarchy map

* **Primary Discovery:** Send Request to <https://www.rei.com/categories> to retrieve JSON response containing the full category structure
* **Data Structure:** Extract parent categories with their nested child categories, preserving the hierarchical relationship
* **Output:** Generate a category tree with canonical URLs (e.g., /c/camping-and-hiking, /c/mens-footwear)

**Phase 2: Product ID Collection via each Category JSON APIs**

**Objective:** Efficiently gather all unique product identifiers

* **API Endpoint Pattern:** [https://www.rei.com/c/{category-slug}?json=true&page={page\_number}](https://www.rei.com/c/%7bcategory-slug%7d?json=true&page=%7bpage_number%7d)
* **Pagination Handling:**
  + Start with page=1 for each category
  + Continue incrementing until receiving empty results or hitting the last page indicator
  + Extract product IDs from JSON response
* **ID Management:** Maintain a centralized set of unique product IDs to avoid duplicate processing
* **Rate Limiting:** Implement delays between requests (3-5 seconds) to respect server resources

**Phase 3: Product Detail Extraction**

**Objective:** Capture comprehensive product information

* **URL Pattern:** Send Request to [https://www.rei.com/product/{product\_id}](https://www.rei.com/product/%7bproduct_id%7d) to retrieve JSON response containing the Product data.
* **Output:** JSON or CSV file with all the desired Product data

**Technical Implementation Considerations**

**Concurrency & Performance:**

* Implement multi-threading with 5-10 concurrent workers for product detail pages
* Use rotating proxies to avoid blockage
* Store Product ids to minimize redundant API calls

**Error Handling & Resilience:**

* Retry failed requests with exponential backoff
* Log failed product IDs for later reprocessing
* Checkpoint progress to enable resume capability