API List for Project Zomboid Modding Tool

**Python Libraries:**

* **os**  
  **Purpose**: To interact with the file system, manage directories, and handle file creation and modification for the modding setup.  
  **Usage**: Automating the setup of the mod project structure, including creating folders and files.
* **json**  
  **Purpose**: To parse and generate JSON data, especially useful for handling mod configuration files and metadata.  
  **Usage**: Reading and writing mod configuration (such as items and recipes) in JSON format.
* **subprocess**  
  **Purpose**: To call and execute shell commands (including Bash and Batch scripts) from within Python.  
  **Usage**: Integrating Python with Bash and Batch scripts to perform various tasks such as file manipulation and system calls.
* **Tkinter**  
  **Purpose**: GUI framework for creating a user-friendly interface for the modding tool.  
  **Usage**: Enabling users to interact with the tool easily through graphical controls like buttons, forms, and text fields.
* **argparse**  
  **Purpose**: To handle command-line arguments and options for the tool.  
  **Usage**: Allowing users to run the modding tool from the command line and pass options for specific tasks (like generating item templates or batch processing files).
* **logging**  
  **Purpose**: To provide a flexible logging system for tracking events, errors, and the overall operation of the modding tool.

**2. Bash and Batch Scripts:**

* **Bash**  
  **Purpose**: A scripting language for automating tasks on Linux and macOS.  
  **Usage**: Automating repetitive modding setup tasks on Linux, such as creating project folders, moving files, and initializing templates for mods.
* **Batch**  
  **Purpose**: A scripting language for automating tasks on Windows.  
  **Usage**: Similar to Bash scripts, but for automating setup tasks on Windows machines. Ensuring compatibility between operating systems for the modding tool.

**3. Cross-Platform Integration:**

* **Platform-Specific Checks** (via Python platform library)  
  **Purpose**: To detect the operating system (Windows vs Linux) and ensure the correct execution of Bash or Batch scripts based on the user’s platform.  
  **Usage**: Allowing your tool to function seamlessly across platforms by executing the right commands depending on whether the tool is running on Windows or Linux.
* **shutil**  
  **Purpose**: To perform high-level file operations in Python (copy, move, rename).  
  **Usage**: Moving mod files and templates around automatically, handling cross-platform file system compatibility.