**RESTful Service Checklist**  **2023-03-15**

* **REST Interface**
  + Are [best practices](https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design) followed in the design of the REST interface endpoints?
  + Are the appropriate REST "verbs" (e.g., PUT, GET, POST, DELETE, and PATCH) used for each action?
  + Are GET, PUT, PATCH, and DELETE endpoints [idempotent](https://restfulapi.net/idempotent-rest-apis/)?
  + Is the REST interface [versioned](https://restfulapi.net/versioning/)?
  + Are the appropriate [HTTP response codes](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status) used?
  + Does the interface at least meet level 2 of the [Richardson maturity model](https://restfulapi.net/richardson-maturity-model/)?
* **Code**
  + Does the implementation satisfy any provided requirements (e.g., in an assignment description)?
  + Are inputs and outputs validated (e.g., using a library)?
  + Did you employ defensive programming techniques to proactively identify and handle errors / unexpected behaviors?
  + Does the internal organization of the software follow best practices (e.g., loosely-coupled components with single responsibilities, depending on interfaces rather than concrete implementations)?
  + Are resources closed and cleaned up after use (e.g., database connections are closed)?
* **Monitoring and Alerting**
  + Is some sort of logging strategy employed? If so, is [structured logging](https://www.structlog.org/en/stable/) used?
  + Are any metrics captured? If so, is an [endpoint available](https://dev.to/camptocamp-ops/implement-prometheus-metrics-in-a-flask-application-p18) for retrieving metrics?
  + Is a [health check endpoint](https://www.ibm.com/garage/method/practices/manage/health-check-apis/) implemented? E.g., testing database connectivity?
* **Security**
  + Are all secrets (e.g., usernames and passwords) passed via environmental variables to avoid accidentally committing files containing secrets to a repository or distributing them with release artifacts?
* **Documentation**
  + Is there a README.md in the GitHub repository?
  + Are there instructions for running the service, including installing any dependencies, setting up a database, etc.?
  + Are there instructions for the process for contributing changes to the repository?
  + Are there instructions for the release process (if appropriate)?
  + Are there instructions for running provided tests?
  + Are there instructions for contacting the project or community members (if appropriate)?
  + Are there instructions for how to report bugs (e.g., submit issues)?
  + If a database is used, is the schema documented? Is a diagram provided if sufficiently complex?
  + Is the internal architecture (e.g., major components / modules and their relationships) documented?
  + Are the REST endpoints documented, including the action type, expected input, and example output (e.g., following [templates](https://github.com/jamescooke/restapidocs))?
* **Bug Reports**
  + Is a template for filing bug reports provided?
* **Automated Tests**
  + Are automated smoke tests provided?
  + Is there a test for each endpoint?
* **License**
  + If the repository is public, is a license specified for the code?