Coding conventions

- Sepparate functions according to their functionally. Avoid one big function.
- Make use of Cloud functions chaining to connect multiple actions.
- Use TypeScript to enforce type definition. Avoid using JavaScript.
- Project is divided into the following folders:
    - *functions/src/endpoints*: Define the interaction between user/system and the API. Open to the exterior.
    - *functions/src/logic*: Defines the logic that will be triggered once an endpoint is reached. Interaction between them it is possible to reach a final goal.
    - *functions/src/models*: Define the entities that are being use in the system.
    - *functions/src/utils*: Define some helper functions that can be reused along the application.
    - *functions/tests*: Define tests to assure code quality of the functions implemented in some of the above folders.

Naming conventions

- Files and folders will be named using underscores and words will be in lower case (e.g. "folder_name" or "file_name.extension")
- To adhere to Pharox styling, an ESLint is present containing all the rules they currently use.