🡪 Agile section

**User story 1:** As a vanilla git power-user that has never seen GiggleGit before, I want to be able to view the published work of my git power-user colleagues.

**User story 2:** As a team lead onboarding an experienced GiggleGit user, I want to keep a safe archive of my old repositories and be able to access them without affecting my current work.

**My own user story:**

As a vanilla git power-user that has never seen GiggleGit before, I want to be able to export my files to users who are not using GiggleGit

* Task: Integrate Export API into GiggleGit
  + Ticket #1: Validate dataflow (make sure what has been sent is the same as what has been received)
  + Ticket #2: Create API keys for all users who want to export

**Why isn’t this a user story?** *As a user I want to be able to authenticate on a new machine*

This isn’t a user story because our epic deals with the onboarding new and exciting clients, not system functionalities and capabilities. Another reason as to why this isn’t a user story is because it states a desired and explicit functionality, user stories aren’t supposed to do this.

🡪 Formal Requirements

**Goal**: Create an easy to use web-interface that lists all the GiggleGit packages clients can use.

**Non-goal:** refresh catalogue of memes every day to avoid user boredom

Non-functional requirement #1: Security

* FR1: Default users should not have same access permissions as PMs.
* FR2: All users (default or PM) should be able to remove/uninstall packages they no longer need as way of closing backdoors.

Non-functional requirement #2: Portability

* FR1: default users as well as PMs should be able to log into GiggleGit regardless of what device they use.
* FR2: Tablet and phone versions of GiggleGit should be designed to be just as useable as laptop/desktop versions. Users (and PMs) should not be confused when using these versions of GiggleGit