

3a Entrega ASW

Abel Batalla
Abdelrahim Chelh
Gabriel del Valle
Arnau Gracia

Canvis realitzats a l'API

- **Obtenir les issues watched d'un usuari:** Hem afegit una crida GET amb paràmetres id d'usuari a `users/{usuari_id}/watchers` per a obtenir les issues que estan watched by l'usuari indicat. Quan es fa un GET de l'usuari no es passen les seves watched issues. Necessiàvem obtenir les watched issues d'un usuari per a poder mostrar-les a la secció de watched issues quan es visualitza un usuari. Vam optar per aquesta forma d'obtenir les watched issues, ja que era una forma molt semblant a la forma d'obtenir les activitats d'un usuari i els canvis al controlador eren mínims.
El nou codi de l'API és el següent:

`/users/{usuari_id}/watchers:`

get:

tags:

- User -> Watchers

summary: Finds watched issues by its UserID

description: Returns all the issues watched by the specified user.

operationId: get_watchers_user

parameters:

- name: usuari_id

in: path

description: ID of user to search

required: true

schema:

type: integer

format: int64

responses:

200:

description: Successful response

content:

application/json:

schema:

type: array

items:

type: object

properties:

id:

type: integer

issue_id:

type: integer

user_id:

type: integer

example:

- id: 113

issue_id: 303

user_id: 2

- id: 122

issue_id: 305

user_id: 2

404:

```
description: 'Error: Not Found'
content:
  application/json:
    schema:
      $ref: '#/components/schemas/ErrorResponse'
    examples:
      errorExample:
        $ref: '#/components/examples/error404'
```

- **Canvi a la funció de filtrar issues:** Antigament, per obtenir les issues filtrades fèiem mitjançant un mateix array la cerca dels diferents filtres dels diferents paràmetres i això no és molt recomanable ni intuïtiu de cara a l'usuari. Amb el canvi realitzat, diferenciem el filtratge segons el paràmetre el qual filtrem i no ajuntem tot en un mateix array. Tal com podem visualitzar a continuació, veurem que hi haurà un array per cada atribut: severity, type, priority, status i assign. El nou codi de l'API és el següent:

```
/issues:
  get:
    tags:
      - Issues
    summary: Get all the issues or a list of filtered, ordered or searched issues
    operationId: index
    parameters:
      - in: query
        name: direction
        schema:
          type: string
        description: Direction to sort the issues by (asc or desc)
      - in: query
        name: order_by
        schema:
          type: string
        description: Attribute to sort the issues by (severity, issue_type, priority, assign, status)
      - in: query
        name: options[]
        schema:
          type: array
          items:
            type: string
      - in: query
        name: severity[]
        schema:
          type: array
          items:
            type: string
      - in: query
        name: type[]
        schema:
          type: array
```

```

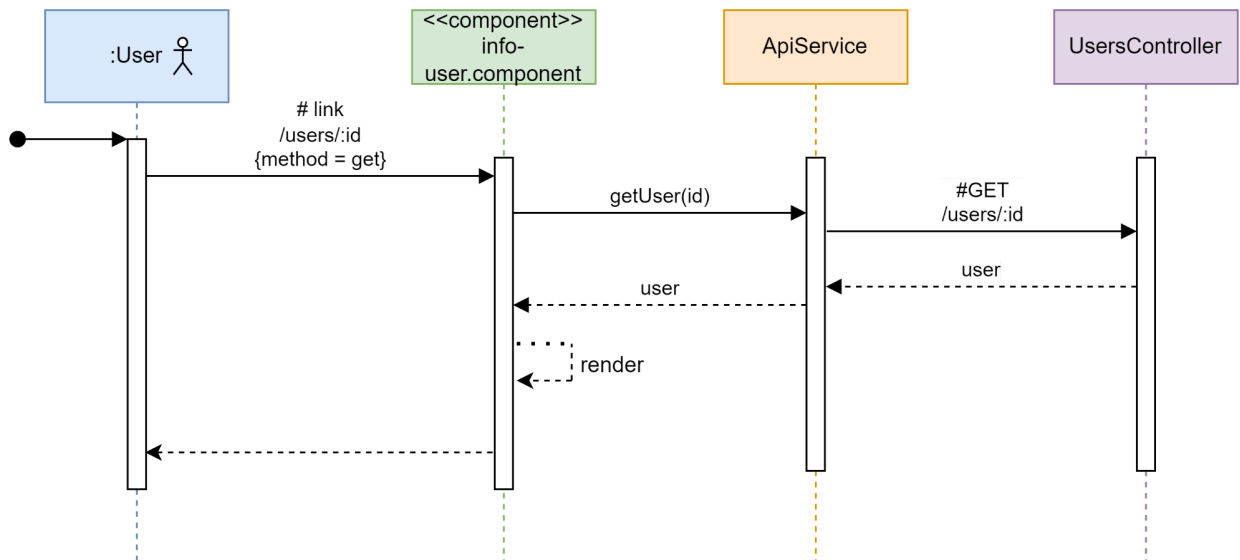
      items:
        type: string
- in: query
  name: priority[]
  schema:
    type: array
    items:
      type: string
- in: query
  name: status[]
  schema:
    type: array
    items:
      type: string
- in: query
  name: assign[]
  schema:
    type: array
    items:
      type: string
description: List of options of Type, Severity, Priority, Subject, Status and Assign to filter
by
- in: query
  name: filtro
  schema:
    type: string
description: Filter to search issues by subject or description

responses:
  200:
    description: List of issues
    content:
      application/json:
        schema:
          type: array
          items:
            $ref: '#/components/schemas/Issue'
  400:
    description: "Error: Bad Request"
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/ErrorResponse'
    examples:
      errorExample:
        $ref: '#/components/examples/error400'

```

Diagrames de seqüència

Clicar l'enllaç per visualitzar la informació de l'usuari "loguejat":



Des de la pantalla anterior, canviar l'about de l'usuari i clicar "update":

