

Investigate status label in the Notifications Queue (LPS-159274)

Objective: investigate the effort to add a new label type on the status column of the Notifications Queue table.

At the time of this spike being made, the status has only two options: sent or unsent. After the story [LPS-159052](#), the expected behavior is to have three different status with their respective colors: sent (**success**), unsent (**warning**) and failed (**error**).

The current implementation of the status logic inside the file

`NotificationQueueEntryStatusDataRenderer.tsx` only allows the use of two labels for the status. The function receives a value (a boolean) to indicate if the email was sent or is unsent.

This logic will have to change to support the three labels, similar to what we have on `ObjectActionStatusDataRenderer.tsx`. In this case we would need to change the value that the function receives from boolean to a object or a string representing the status to be set; it needs to be discussed with the backend responsible if the properties `label` e `label_i18n` will be necessary in this case or it will be another type of structure. Also probably it will be necessary to create a file similar to the `ObjectActionConstants.java` regarding the three status mentioned before.

I assume that the frontend work would be minimum in this case (considering that the value sent to the function will be an object), changing the behavior of the

`NotificationQueueEntryStatusDataRenderer.tsx` logic into something like this:

```
export function NotificationQueueEntryStatusDataRenderer({value}: IProps) {

  const getStatusCssClass = (label: string) => {
    switch (label) {
      case 'failed':
        return 'label-danger';
      case 'unsent':
        return 'label-warning';
      case 'sent':
        return 'label-success';
      default:
        return '';
    }
  };

  return value ? (
    <strong className={`label ${getStatusCssClass(value.label)} `}>
      {value.label_i18n}
    </strong>
  ) : null;
}
```

```
        </strong>
    ) : null;
}

interface IProps {
    value: {
        label: string;
        label_i18n: string;
    };
}
```

For more information about the implementation of a similar status on the Action side, you can see this [PR](#). The current implementation of the Notification Queue status can be found [here](#).