

# FANTASTIC LABELS

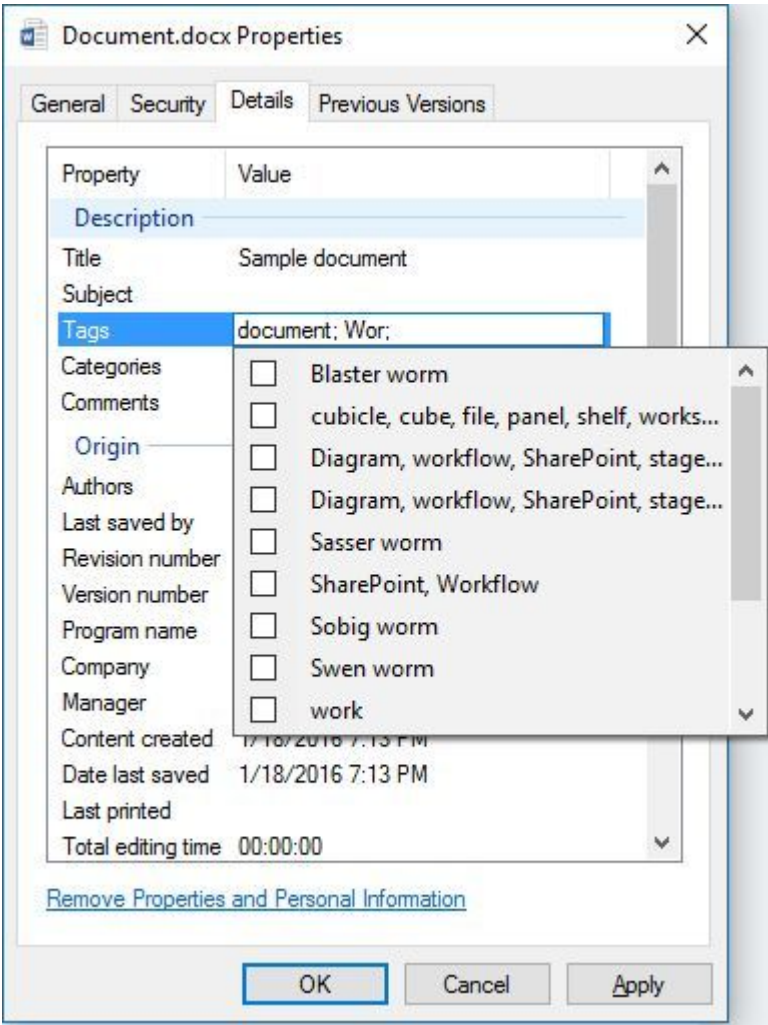
AND WHERE  
TO FIND THEM



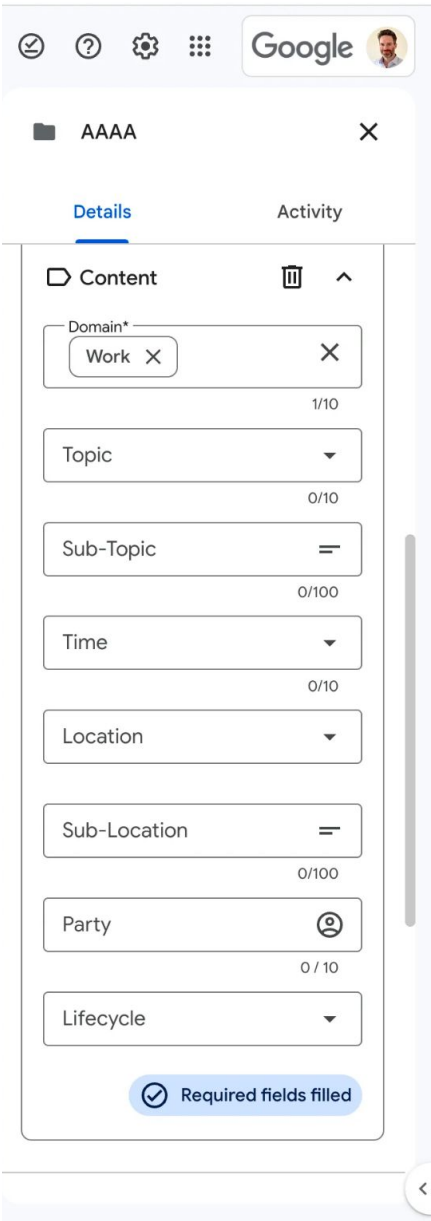
In the quest of an optimized research in Google Drive, many teams asked me to place some metadata on their documents and files.

Widespread in Microsoft environment, this tool is not really known in Google Drive by users.

### Microsoft



### Google Drive



As Google Workspace admin or may I say wizards, we can use some little tricks to answer teams requirements.

First, we can create labels with the team : [Create label in the admin console](#)

The label can have different fields like list, number, text,..

← Sensibilité Publié, modifications enregistrées en tant que brouillon

Autorisations Publier les modifications

Modifier un libellé Règles

Nom du libellé\*  
Sensibilité

Description du libellé  
Vérifie la sensibilité du document

Ajouter un lien "En savoir plus"

En cas de copie des fichiers  
Toujours copier le libellé

Options

☒ Demander aux utilisateurs de choisir une option

Remarque : La réorganisation des options est limitée pour ce champ, car certaines options sont associées à une règle

1.	Administratif interne	Ne pas partager	
2.	Administratif externe	A partager avec précaution	
3.	Client	Partagé avec un accès restreint au client	
4.	Public	Partage en public	
5.	Technique		

Ajouter une option

Aperçu

Brouillon Publiée

Sensibilité  
Vérifie la sensibilité du document

ⓘ Ce libellé comporte des champs obligatoires

Autorisations Modifier

État	Publié, modifications enregistrées en tant que brouillon
Concepteur	Ghislain SANJUAN
Dernière modification	Aujourd'hui • Ghislain SANJUAN


You can also apply some DLP rules if you want !






Once the label is created, it needs to be secure and visible by the right people.

An invisibility cloak can be used by setting up some parameters like who can edit or see the label.



Autorisations liées aux libellés



Sango


Permet de consulter ce libellé ▼

Tous les utilisateurs peuvent afficher et rechercher ce libellé pour les fichiers qu'ils sont autorisés à afficher ou commenter

Autorisations avancées

Les utilisateurs ou les groupes ajoutés peuvent utiliser le libellé

Ajouter des personnes ou des groupes



Ghislain SANJUAN  
ghislain.sanjuan@sango-co.com

Permet d'appliquer des libellés et de définir leur valeur ▼

Remarque : Les modifications d'autorisation ne nécessitent pas la publication du libellé

Annuler

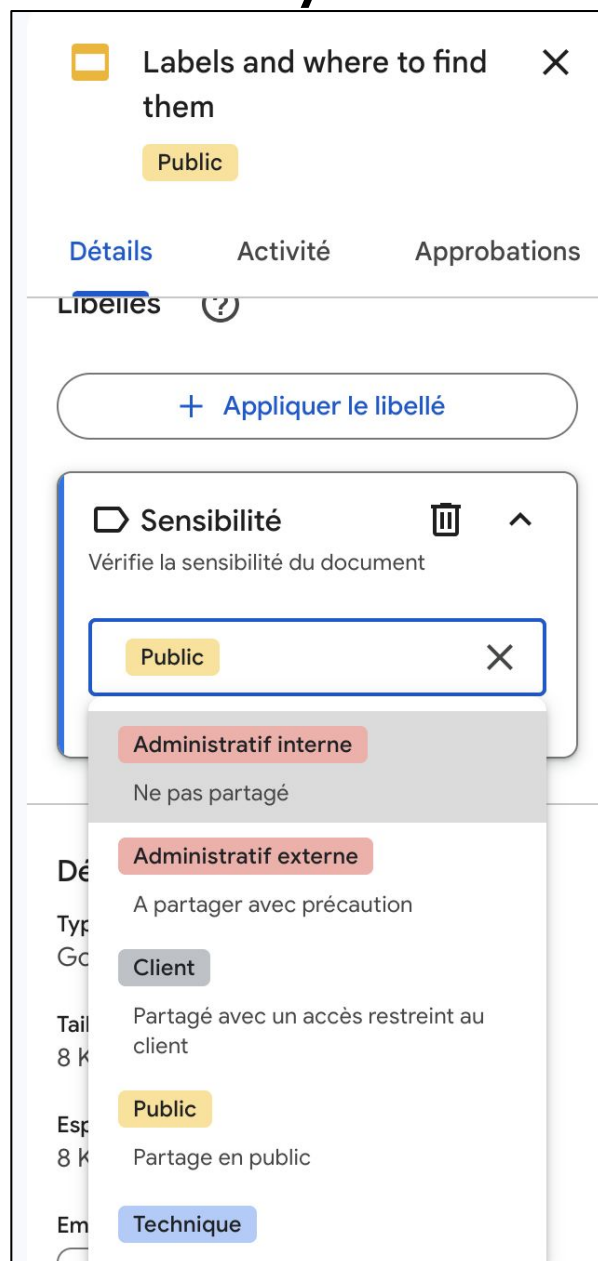
Enregistrer



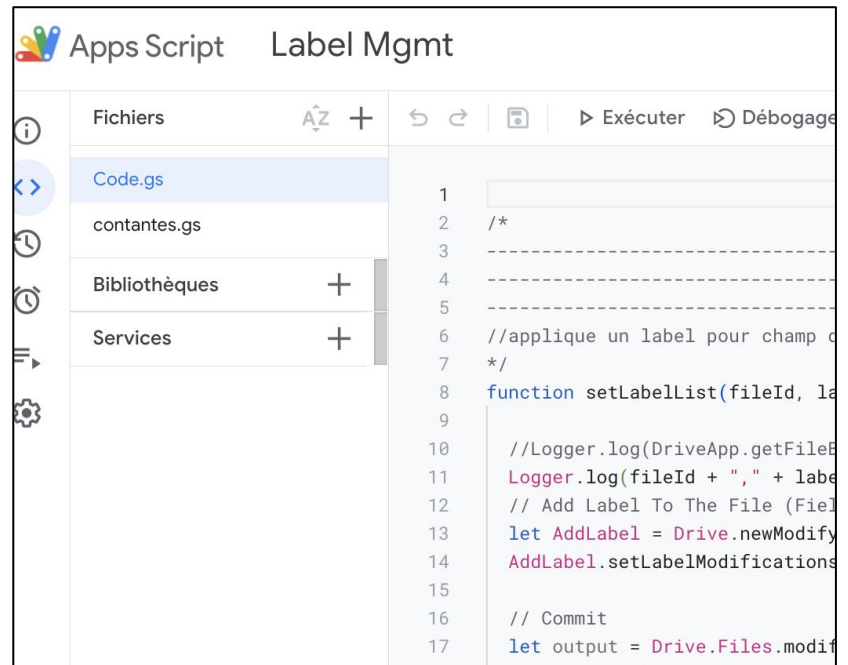
Once the spell the label is ready, we need to apply it on every files the user/team wants.

2 options are available :

Manually apply the label on every item if not too many files



If a lot of files to manage, you can use a little magic with some script

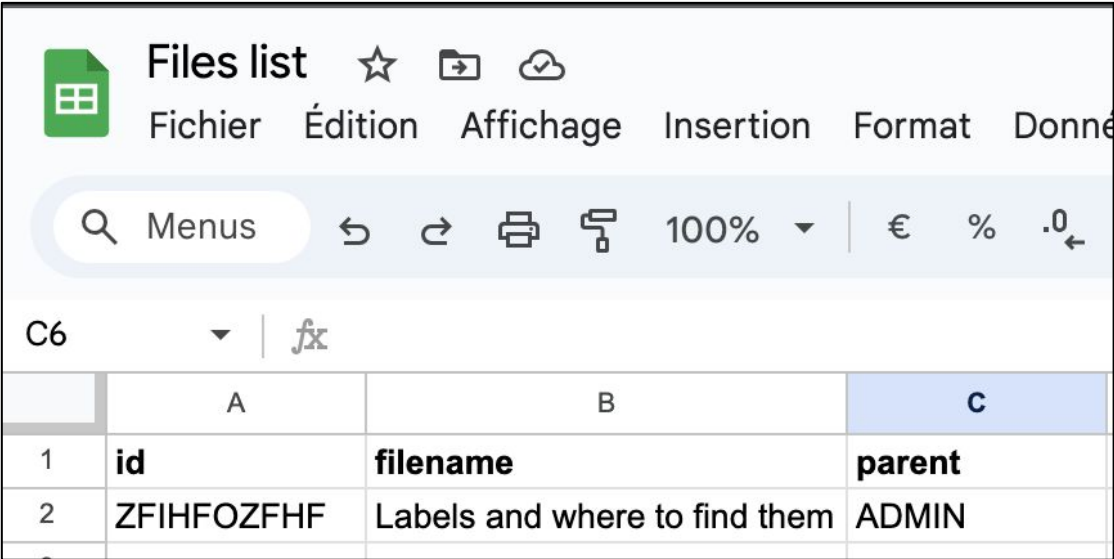


The script option needs first to extract in a Google Sheets the files list in the folder where the items are. You can use GAM to do so with this command

```
gam user ghislain.sanjuan@sango-co.com show filelist
select id folder_ID fields id,title,parents todrive
tduser ghislain.sanjuan.externe@sango-co.com
```

Then each line/item needs to be reviewed by the end user in order to know what label value needs to be applied.

This step can be helped by gathering the folder or file name : if the *parent folder name* contains 2000 , then apply the label *Archive* for exemple.



Files list			
	Fichier	Édition	Affichage
	Insertion	Format	Donnée
	Menus	100%	€ % .0
C6	A	B	C
1	id	filename	parent
2	ZFIHFOZFHf	Labels and where to find them	ADMIN



**In the meantime, you can check out the label structure created with this script**

```
/*
/Analyse the label and extract its informations (fields ids, options ids,...)
*/
function getLabelInformation() {
  //Place label id -> Example https://drive.google.com/labels/v2793JJ
  let label_json = DriveLabels.Labels.get("labels/v2793JJ", {
    "view": "LABEL_VIEW_FULL"
  })
  label_json.fields.forEach(function (t) {
    let text
    try {
      text = `"id": "` + t.id + `",
      "options": [`
      t.selectionOptions.choices.forEach(function (c) {
        text += `{
          "name": "` + c.properties.displayName + `",
          "id": "` + c.id + `",
        },`
      })
    }
    catch (e) {
      Logger.log(e)
      text = t.properties.displayName + ":" + t.id
    }
    console.log(text)
    return text
  })
}
```

**This will give you the label fields IDs and other informations (names, options). It's basically the Label JSON format.**



**The Label JSON format is created and the files list is set up with the labels value to apply for each file ?**

**If yes, you can cast the spell that will apply the right value to the right label filed for every item.**

**To do so, you can use Apps Script to apply the value to the different label fields.**

```

/*****
 * This function loop over the files list and apply the right value for the label filed
 * *****/
@param {}
@return {}
*/
function setlabels() {
  let data = SP.getSheetByName("XXXXX").getDataRange().getValues()
  //If the script is too lng (because the files list is too long ),
  //We can break down the function call by breaking the data into smaller array of 500 lines
  //That's why we add a line number, to know where we stopped and to resume on the next call
  let index = Number(PropertiesService.getScriptProperties().getProperty("startIndex"))
  for (let i = index; i < index + step; i++) {
    let docId = data[i][xx] //On récupère l'id du fichier pour lequel on va appliquer le label
    try {
      //We gather the value to add on the label fields
      let champ1Valeur = data[i][xx]
      let champ2Valeur = data[i][xx]
      let champ3Valeur = data[i][xx]
      //We apply a filter for the list type label fields using the value set up in the Google Sheets/files list
      /*
      {
        "name": "bordeaux",
        "id": "xxxxxx"
      }
      */
      let champ4Valeur = champ4Liste.options.filter(x => x.name == data[i][xx])[0].id

      //We apply the values to the label fields using functions depending on the field type
      setLabelText(docId, LABEL.id, LABEL.champ1.id, champ1Valeur)
      setLabelText(docId, LABEL.id, LABEL.champ2.id, champ2Valeur)
      setLabelInteger(docId, LABEL.id, LABEL.champ3.id, champ3Valeur)
      setLabelList(docId, LABEL.id, LABEL.champ4.id, champ4Valeur)
    }
    catch (e) {
      console.log(e + ":" + i + ":" + data[i][2])
    }
  }
  //Index incrementation
  PropertiesService.getScriptProperties().setProperty("startIndex", index + step)
}
```





# Want to discover the entire code ?

## Check out the GitHub project

[Fantastics labels and where to find them](#)

