

DC Labs

Guide pratique — Publier votre outil

Pour tous les profils, même sans expérience technique

💡 **Ce guide explique tout pas à pas.** Chaque mot technique est expliqué dès sa première apparition. Vous n'avez besoin d'aucune connaissance préalable pour suivre ce guide.

Avant de commencer — comprendre le principe

Comment fonctionne la publication d'un outil ?

Imaginez DC Labs comme un tableau d'affichage commun pour toute l'équipe. Quand vous avez créé un outil (une petite application web), vous devez l'afficher sur ce tableau pour que tout le monde puisse l'utiliser.

Voici ce qui se passe concrètement, de façon simplifiée :

1. Vous déposez les fichiers de votre outil dans un espace partagé sur internet (**GitHub**, voir définition ci-dessous)
2. Un robot automatique détecte votre dépôt et publie votre outil sur le site DC Labs en **1 à 3 minutes**
3. Votre outil est maintenant accessible à toute l'équipe via le portail

✅ **Ce que vous n'aurez jamais à faire :** vous connecter au serveur IONOS, manipuler des fichiers serveur, ou demander à quelqu'un de publier à votre place.

Glossaire — les mots à connaître

Voici les termes que vous allez rencontrer dans ce guide, expliqués simplement :

🔍 **GitHub :** *Un site web (github.com) qui permet de stocker et partager des fichiers de code. C'est comme Google Drive, mais spécialisé pour les développeurs. DC Labs utilise GitHub pour centraliser tous les fichiers du portail.*

🔍 **Dépôt (ou "repo") :** *Un dossier de projet stocké sur GitHub. Le dépôt DC Labs s'appelle FactoryDianeIA/dc-labs et contient tous les fichiers du portail et de ses outils.*

🔍 **Git :** *Un programme à installer sur votre ordinateur qui vous permet d'envoyer vos fichiers vers GitHub. C'est le "pont" entre votre machine et GitHub.*

🔍 **Terminal / Invite de commandes :** *Une fenêtre noire où on tape des instructions textuelles à l'ordinateur. Sur Mac : cherchez "Terminal" dans vos applications. Sur Windows : cherchez "Invite de commandes" ou "PowerShell".*

🔍 **Cloner :** *Télécharger une copie complète du dépôt GitHub sur votre ordinateur. À faire une seule fois.*

🔍 **Commit :** *Enregistrer officiellement vos modifications avec un message qui décrit ce que vous avez fait. Comme un "point de sauvegarde" avec une étiquette.*

- 🔍 **Push** : Envoyer vos modifications depuis votre ordinateur vers GitHub, pour les rendre visibles par tous.
- 🔍 **Pull** : Récupérer les dernières modifications faites par vos collègues depuis GitHub vers votre ordinateur. À toujours faire avant de commencer à travailler.
- 🔍 **Branch / Branche main** : La version principale et officielle du projet. Tout ce qui est publié sur le site passe par la branche "main".
- 🔍 **HTML/CSS/JS** : Les langages utilisés pour créer des pages web. HTML = la structure, CSS = le style visuel, JavaScript (JS) = les interactions. Votre outil doit être fait dans ces langages.
- 🔍 **index.html** : Le fichier principal d'une page web. Quand on ouvre un dossier de site, c'est ce fichier qui s'affiche en premier.
- 🔍 **tools.json** : Un fichier texte qui contient la liste de tous les outils du portail. C'est la "table des matières" du site DC Labs. Vous devrez y ajouter une ligne pour enregistrer votre outil.
- 🔍 **JSON** : Un format de texte structuré très utilisé pour stocker des données. Facile à lire pour les humains et les ordinateurs. Chaque outil du portail est décrit par un bloc JSON dans tools.json.
- 🔍 **Kebab-case** : Une convention de nommage : mots en minuscules séparés par des tirets. Exemple : mon-super-outil. À utiliser pour nommer le dossier de votre outil.
- 🔍 **localhost:8080** : Une adresse web qui ne fonctionne que sur votre ordinateur. Permet de tester votre outil avant de le publier. '8080' est le numéro du port (comme un numéro de porte dans un immeuble).
- 🔍 **CI / GitHub Actions** : Un robot automatique configuré dans le projet. Dès que vous envoyez des fichiers sur GitHub (push), ce robot publie automatiquement les changements sur le vrai site DC Labs.

Prérequis — ce qu'il faut avoir avant de commencer

Avant de pouvoir publier un outil, vous avez besoin de trois choses. Voici comment les obtenir si vous ne les avez pas encore :

1. Un compte GitHub

GitHub est gratuit. Si vous n'en avez pas encore :

4. Allez sur **github.com** dans votre navigateur
5. Cliquez sur "Sign up" (s'inscrire) en haut à droite
6. Remplissez le formulaire avec votre adresse email professionnelle
7. Une fois inscrit, envoyez votre nom d'utilisateur GitHub à l'administrateur (FactoryDianeIA) pour être ajouté au dépôt

⚠️ **Important** : sans être ajouté comme collaborateur par l'administrateur, vous ne pourrez pas envoyer vos fichiers. Cette étape est obligatoire.

2. Git installé sur votre ordinateur

Git est le programme qui fait le lien entre votre ordinateur et GitHub.

Pour vérifier si Git est déjà installé : ouvrez votre Terminal (Mac) ou Invite de commandes (Windows), tapez la commande ci-dessous et appuyez sur Entrée :

```
Git --version
```

Si vous voyez un numéro de version (ex : git version 2.39.0), Git est déjà installé. Sinon :

8. Allez sur git-scm.com et téléchargez la version pour votre système (Windows ou Mac)
9. Installez-le en suivant les étapes (laissez toutes les options par défaut)
10. Relancez votre Terminal et retapez git --version pour confirmer


3. Avoir cloné le projet (une seule fois)

Cloner = télécharger une copie du projet sur votre ordinateur. Cette étape n'est à faire qu'une seule fois.

Ouvrez votre Terminal, choisissez où vous voulez copier le dossier et tapez ces deux commandes (appuyez sur Entrée après chaque ligne) :

```
git clone https://github.com/FactoryDianaA/dc-labs.git  
cd dc-labs
```

Que font ces commandes ? La première télécharge tout le projet dans un dossier "dc-labs" sur votre ordinateur. La seconde vous "déplace" dans ce dossier pour travailler dedans.

 **Astuce :** le dossier "dc-labs" sera créé là où votre Terminal est ouvert (généralement votre dossier Utilisateur). Vous pouvez le retrouver dans votre explorateur de fichiers comme n'importe quel dossier.

Publier votre outil — les 6 étapes

 **Vue d'ensemble :** mettre à jour → créer votre dossier → s'inscrire dans la liste → tester → envoyer → vérifier en ligne.

ÉTAPE 1 Mettre à jour votre copie locale

Pourquoi c'est important : depuis la dernière fois que vous avez utilisé le projet, vos collègues ont peut-être ajouté des outils. Si vous ne récupérez pas leurs changements avant de travailler, vous risquez des conflits (deux personnes qui modifient le même fichier en même temps).

Ouvrez votre Terminal, assurez-vous d'être dans le dossier dc-labs, puis tapez :

```
git checkout main
```

```
git pull
```

Que font ces commandes ? La première s'assure que vous travaillez sur la version principale (main). La seconde télécharge les dernières modifications de vos collègues depuis GitHub.

💡 **Conseil** : prenez l'habitude de toujours faire ces deux commandes avant de commencer à travailler. Ça évite 99% des problèmes.

ÉTAPE 2 Créer le dossier de votre outil

Chaque outil a son propre dossier dans `mon-site/tools/`. Ce dossier doit contenir au minimum un fichier `index.html` (la page principale de votre outil).

Comment nommer votre dossier ?

Utilisez le format **kebab-case** : tout en minuscules, les mots séparés par des tirets, sans espaces ni caractères spéciaux.

✅ Correct	❌ Incorrect
convertisseur-csv	Convertisseur CSV
generateur-rapports	générRateur_Rapports
outil-rh-2026	outil rh 2026

Structure attendue de votre dossier

Voici à quoi doit ressembler votre dossier une fois créé (exemple avec "convertisseur-csv") :

```
mon-site/tools/convertisseur-csv/
├─ index.html      ← obligatoire : page principale de l'outil
├─ style.css       ← optionnel : fichier de style
├─ script.js       ← optionnel : fichier JavaScript
└─ autres-fichiers ← optionnel : images, données...
```

Règles importantes à respecter :

Règle	Explication
Un fichier <code>index.html</code> obligatoire	C'est la porte d'entrée de votre outil. Sans lui, le site ne sait pas quoi afficher quand on clique sur votre carte.

Règle	Explication
Fichiers statiques uniquement	Votre outil doit fonctionner avec HTML, CSS et JavaScript uniquement. Pas de serveur Node.js ou PHP qui doit "tourner" en permanence.
Ajouter un lien Retour	Mettez un lien en haut de votre page qui ramène au portail : <code>← Retour au portail</code>
Nom de dossier unique	Vérifiez dans <code>mon-site/tools/</code> qu'aucun dossier n'a déjà ce nom.

ÉTAPE 3 S'inscrire dans la liste des outils (tools.json)

Qu'est-ce que tools.json ? C'est le fichier qui contient la liste de tous les outils du portail. Quand vous ajoutez votre entrée ici, le portail affiche automatiquement votre carte sur la page d'accueil.

Ce fichier se trouve ici : `mon-site/tools.json`

Ouvrez-le avec un éditeur de texte (Notepad, VS Code, TextEdit...) et ajoutez le bloc suivant à la fin de la liste, avant le] final :

```
{
  "nom": "Convertisseur CSV",
  "auteur": "Prénom Nom",
  "dossier": "convertisseur-csv",
  "url": "",
  "icone": "🇫🇷",
  "description": "Convertit un fichier CSV en tableau HTML.",
  "tags": ["csv", "data"],
  "date_ajout": "2026-06-12"
}
```

Que signifie chaque champ ?

Champ	Ce qu'il faut mettre	Exemple
nom	Le nom de votre outil, tel qu'il apparaîtra sur la carte	Convertisseur CSV
auteur	Votre prénom et votre nom	Marie Dupont
dossier	Le nom exact du dossier que vous avez créé à l'étape 2	convertisseur-csv
url	Laissez vide "" si votre outil est dans le dossier tools/	""
icone	Un emoji qui représente votre outil	🇫🇷 ou 🛠️ ou 📄
description	Une phrase courte qui explique ce que fait l'outil	Convertit un CSV en tableau
tags	Des mots-clés pour la recherche (en liste)	["csv", "data"]

Champ	Ce qu'il faut mettre	Exemple
date_ajout	La date du jour au format AAAA-MM-JJ	2026-06-12

⚠ **Attention aux virgules** : en JSON, chaque bloc (entre { et }) doit être suivi d'une virgule, SAUF le dernier. Si vous ajoutez votre outil à la fin, assurez-vous qu'il y a une virgule après l'accolade fermante } du bloc précédent, mais pas après le vôtre.

Exemple correct avec deux outils dans le fichier :

```
[
  {
    "nom": "Outil de Marie",
    ...
  },      ← virgule ici car ce n'est pas le dernier
  {
    "nom": "Votre nouvel outil",
    ...
  }      ← pas de virgule ici car c'est le dernier
]
```

ÉTAPE 4 Tester votre outil en local

Pourquoi tester avant de publier ? Pour éviter d'envoyer un outil cassé sur le vrai site. Tester en local signifie vérifier que tout fonctionne sur votre ordinateur avant que tout le monde puisse le voir.

Ouvrez votre Terminal, placez-vous dans le dossier mon-site, puis lancez cette commande :

```
cd mon-site
python3 -m http.server 8080
```

Que fait cette commande ? Elle démarre un mini-serveur web sur votre ordinateur. Votre portail devient accessible à l'adresse <http://localhost:8080> dans votre navigateur.

💡 **Si python3 ne fonctionne pas** : essayez `python -m http.server 8080` (sans le 3). Sur Windows, vous pouvez aussi simplement double-cliquer sur `index.html` pour l'ouvrir dans votre navigateur.

Liste de vérification avant de continuer :

- Ma carte apparaît bien dans la grille du portail
- Un clic sur ma carte ouvre bien mon outil
- Mon outil fonctionne correctement (fonctionnalités, boutons, etc.)
- Le lien « Retour au portail » ramène bien à la page d'accueil
- La recherche et les filtres par tags fonctionnent avec mon outil

Pour arrêter le serveur de test, retournez dans le Terminal et appuyez sur Ctrl + C.

ÉTAPE 5 Envoyer vos fichiers sur GitHub

C'est l'étape de publication. Vous allez envoyer vos fichiers vers GitHub. Le robot automatique (CI) se chargera ensuite de les publier sur le vrai site DC Labs.

Dans le Terminal, depuis la racine du dossier dc-labs (tapez `cd ..` si vous étiez dans mon-site) :

Commande 1 — Sélectionner vos fichiers à envoyer (exemple)

```
git add mon-site/tools/convertisseur-csv/ mon-site/tools.json
```

Que fait cette commande ? Elle dit à Git : "ces fichiers-là, je veux les inclure dans mon prochain envoi". Remplacez `convertisseur-csv` par le vrai nom de votre dossier.

Commande 2 — Créer un point de sauvegarde (commit) (exemple)

git commit -m « ajout outil : convertisseur CSV »

Que fait cette commande ? Elle crée un enregistrement officiel de vos modifications avec un message descriptif. Personnalisez le message entre guillemets avec le nom de votre outil.

Commande 3 — Envoyer vers GitHub (push)


git push origin main

Que fait cette commande ? Elle envoie votre sauvegarde vers GitHub, où le robot automatique va la détecter et publier votre outil sur le vrai site.

ÉTAPE 6 Vérifier que votre outil est en ligne

Après votre push, le déploiement automatique prend entre 1 et 3 minutes. Voici comment suivre son avancement :

11. Allez sur github.com/FactoryDianeIA/dc-labs
12. Cliquez sur l'onglet **Actions** (en haut du dépôt)
13. Vous verrez une ligne avec le nom de votre commit — attendez qu'une coche verte apparaisse
14. Ouvrez le portail DC Labs dans votre navigateur et vérifiez que votre carte est bien là

Ce que vous voyez dans Actions	Signification
 Cercle orange qui tourne	Déploiement en cours — patientez 1 à 2 minutes

Ce que vous voyez dans Actions	Signification
✓ Coche verte	Déploiement réussi — votre outil est en ligne !
✗ Croix rouge	Une erreur s'est produite — voir la section Dépannage

Alternative — publier sans utiliser le Terminal

Si le Terminal vous intimide, vous pouvez tout faire directement depuis le site GitHub dans votre navigateur. C'est un peu moins rapide, mais tout aussi efficace.

15. Allez sur github.com/FactoryDianeIA/dc-labs dans votre navigateur
16. Naviguez vers le dossier `mon-site/tools/`
17. Cliquez sur **Add file** → **Upload files** et glissez-déposez votre dossier entier
18. Revenez à la racine du dépôt, ouvrez `mon-site/tools.json` et cliquez sur l'icône crayon (✎) pour l'éditer
19. Ajoutez votre entrée JSON (voir Étape 3), puis cliquez sur **Commit changes**

💡 **Le déploiement automatique se déclenche exactement de la même façon qu'avec le Terminal. Vous verrez la même coche verte dans l'onglet Actions.**

Cas particulier — votre outil est déjà en ligne ailleurs

Si votre outil est déjà hébergé sur un autre service (comme Vercel, Netlify, ou un autre hébergeur), vous n'avez pas besoin de créer un dossier dans `tools/`. Il suffit d'ajouter une ligne dans `tools.json` avec l'URL de votre outil :

```
{
  "nom": "Mon Dashboard",
  "auteur": "Prénom Nom",
  "dossier": "",
  "url": "https://mon-outil.vercel.app", ← l'URL de votre outil
  "icone": "📊",
  "description": "Tableau de bord KPI en temps réel.",
  "tags": ["dashboard"],
  "date_ajout": "2026-06-12"
}
```

La différence avec le cas normal : vous laissez `"dossier"` vide et remplissez `"url"` avec l'adresse complète de votre outil.

En cas de problème — solutions aux erreurs courantes

Problème rencontré	Cause probable	Solution
git push refusé / "Permission denied"	Vous n'êtes pas encore collaborateur du dépôt, ou votre connexion GitHub n'est pas configurée	Vérifiez avec l'administrateur que vous avez bien été ajouté. Essayez git pull d'abord.
Conflit sur tools.json	Un collègue a modifié tools.json en même temps que vous	Faites git pull. Git vous montrera les deux versions. Gardez les deux entrées, sauvegardez, puis refaites git push.
Ma carte apparaît en local mais pas sur le vrai site	Le déploiement automatique a peut-être échoué	Vérifiez l'onglet Actions sur GitHub. Une croix rouge indique une erreur — contactez l'administrateur.
Clic sur la carte → page d'erreur 404	Le nom dans "dossier" dans tools.json ne correspond pas exactement au nom du dossier créé	Vérifiez que le champ "dossier" dans tools.json est identique (majuscules, tirets) au nom réel du dossier dans tools/.
"JSON invalide" ou le portail ne charge plus	Une virgule manquante ou en trop dans tools.json	Copiez le contenu de tools.json sur jsonlint.com pour identifier l'erreur. Vérifiez surtout les virgules entre les blocs.
git : command not found	Git n'est pas installé sur votre ordinateur	Installez Git depuis git-scm.com (voir section Prérequis).

Rappel — ce que vous n'avez PAS à faire

Pour éviter toute confusion, voici ce qui n'est jamais nécessaire :

- **Se connecter à IONOS ou utiliser FileZilla** (le déploiement est 100% automatique via GitHub)
- Lancer le script `deploy.sh` manuellement (c'est réservé à l'administrateur en cas d'urgence)
- Modifier le fichier `index.html` du portail principal (seulement tools.json et votre dossier dans tools/)
- **Demander à quelqu'un de publier à votre place** (vous avez tous les droits nécessaires)

? **Besoin d'aide ?** Contactez **AdrienMereghetti-DC** sur GitHub — administrateur du dépôt [FactoryDianaI/dc-labs](https://github.com/FactoryDianaI/dc-labs)