Créer et publier votre premier site

Pour la première journée d'un créateur de sites sur la plateforme Website Strategy, le service d'hébergement qui construit et sert les sites Web des clients. À la fin, `ws` est installé, un site est ouvert sur votre téléphone et un **échantillon** est publié, avec un lien à partager. Un échantillon est un site gratuit fait pour une entreprise qui n'est pas encore cliente. Vous apprenez aussi à répondre à la demande d'un propriétaire. Comptez environ 30 minutes.

Les exemples suivent Jordan Bell, un créateur de sites, qui fait le site de Saltmere Marina. Vos noms, liens et codes seront différents. Les commandes et ce qu'elles affichent restent en anglais, comme `ws` les écrit.

Avant de commencer

Il vous faut :

- **Un compte employé Outport.** Outport est le service de connexion de Sevenview pour ses propres outils, à https://outport.app. Jordan se connecte avec `jordan@example.com`.
- **Git.** `ws` utilise Git pour envoyer et récupérer les fichiers d'un site.
- **Un téléphone ou un deuxième navigateur,** pour ouvrir les aperçus.

1. Installer ws et se connecter

1. Connectez-vous à https://outport.app avec votre compte employé Outport.
2. Ouvrez **Tools > Software**, à https://outport.app/tools/software, et installez `ws` pour macOS ou Linux. Si vous avez déjà l'outil `out` d'Outport, `out upgrade ws` l'installe aussi, et `out upgrade` le garde à jour.
3. Vérifiez qu'il fonctionne :

   ```
   $ ws version
   ws 0.1.56
   ```

   Votre numéro peut être plus élevé.
4. Vérifiez que Git est installé et connaît votre nom et votre courriel, que chaque commit enregistre :

```
$ git --version
$ git config user.name
$ git config user.email
```

Si `git` est introuvable, installez d'abord Git. Si l'une des lignes `git config` n'affiche rien, réglez-la :

```
$ git config --global user.name "Jordan Bell"
$ git config --global user.email jordan@example.com
```

5. Connectez-vous :

```
$ ws login
Opening https://outport.app in your browser to sign in.
```

Connectez-vous avec votre compte employé Outport dans le navigateur. `ws` affiche ensuite le nom avec lequel vous êtes connecté, et reste connecté.
Quand le navigateur est sur un autre ordinateur, par exemple quand `ws` tourne sur un serveur ou par SSH, connectez-vous plutôt avec un code. Un agent de programmation se connecte aussi de cette façon, parce qu'il ne peut pas ouvrir de navigateur :

```
$ ws login --device
On another device, open https://outport.app/oauth/device and enter 55NB-64GH.
Signed in as Jordan Bell.
```

Ouvrez cette adresse dans n'importe quel navigateur, connectez-vous avec votre compte employé Outport et tapez le code. Chaque connexion affiche un nouveau code.

6. Vérifiez l'environnement :

```
$ ws profile show
Environment: production (https://api.websitestrategy.com/api)
Profile: production
API: https://api.websitestrategy.com/api
```

Un `ws` installé utilise `production`, là où vivent les sites des vrais clients, et chaque étape de ce guide s'y fait.

2. Créer un site

Lancez `ws new` avec le nom de l'entreprise :

```
$ ws new "Saltmere Marina"
Made the site "Saltmere Marina".
  Subdomain:     saltmere-marina
  Language:      English (en)
  Time zone:     America/Halifax
  Folder:        ./saltmere-marina (a Git clone, 6 files)

Next: cd saltmere-marina && ws preview
```

`ws new` donne deux noms au site :

- **Le nom affiché,** « Saltmere Marina », est celui que les gens lisent. Les pages l'affichent avec `{{ site.name }}`.
- **Le sous-domaine,** `saltmere-marina`, commence les adresses Web du site sur notre domaine d'aperçu, `sitepeek.page`. Il utilise des lettres minuscules, des chiffres et des traits d'union simples, et deux sites ne peuvent pas avoir le même. Pour le choisir vous-même, ajoutez `--subdomain saltmere`.

La langue et le fuseau horaire viennent de votre ordinateur. Réglez-les sur ceux de l'entreprise : pour une entreprise de Terre-Neuve dont le site Web est en anglais et en français, lancez `ws new "Nom" --languages en,fr --time-zone America/St_Johns`. `ws` ne les cherche pas pour vous.

Le dossier est un clone Git du dépôt du site sur la plateforme, avec un commit qui contient les fichiers de départ :

```
$ cd saltmere-marina
$ ls -A
.git  .gitignore  .ws  AGENTS.md  _includes  _layouts  index.html  styles.css
```

`index.html` est la page d'accueil, à `/`. Chaque page est un fichier HTML nommé d'après son adresse : `about.html` est `/about`. `_layouts/default.html` est le HTML autour de chaque page, et `_includes/` contient l'en-tête et le pied de page que toutes les pages partagent. Ne touchez pas à `.ws/` : `ws` y garde l'identifiant du site. Make a site and preview it présente chaque fichier, en anglais.

Si ce site n'est qu'un essai, faites en sorte que son lien d'échantillon prenne fin un jour après sa dernière publication. Lancez ceci dans le dossier du site :

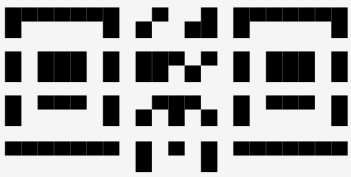
```
$ ws site --expires 1d
The sample's private link now lasts 1 day from each publish.
Saved settings for "Saltmere Marina".
```

Quand ce lien prend fin, la plateforme supprime le site, parce que personne ne l'a acheté.

3. L'ouvrir sur votre téléphone

```
$ ws preview
Built 1 page.

Private preview, open to anyone with this link until November 20:
https://saltmere-marina--k7f3.sitepeek.page/?key=Hn4pX7rKq2Lw9TzB3mVc8d



Scan the code with your phone's camera to open the preview.
```

Lisez le code avec l'appareil photo de votre téléphone, ou ouvrez le lien dans n'importe quel navigateur. Le code ne s'affiche que quand `ws` écrit dans une fenêtre de terminal; le lien est toujours là. Chaque `ws preview` garde le même lien, ouvert 7 jours de plus. La clé à la fin du lien donne à ce navigateur l'accès à cet aperçu seulement, puis elle est retirée de l'adresse. Un navigateur qui n'a jamais ouvert le lien avec sa clé voit seulement « This preview is private ».

Un aperçu n'est jamais publié, et ce n'est pas une copie de sauvegarde : aucune commande n'en récupère les fichiers.

4. Modifier le site

1. Laissez l'aperçu surveiller le dossier, pour que chaque page ouverte se mette à jour quand vous enregistrez :

```
$ ws preview --watch
Built 1 page.
Private preview: https://saltmere-marina--k7f3.sitepeek.page/?
key=Hn4pX7rKq2Lw9TzB3mVc8d
Watching this folder. Press Ctrl+C to stop.
```

2. Dans une autre fenêtre de terminal, ouvrez `index.html`. Voici la page au complet :

   ```
   ---
   layout: default
   title: Home
   ---
   <h1>{{ site.name }}</h1>
   <p>Welcome. This page is index.html.</p>
   ```

   Les lignes entre les lignes `---` sont les réglages de la page. Les balises entre accolades sont du Liquid, le langage de gabarits créé par Shopify.
3. Changez la phrase de bienvenue et enregistrez le fichier. La page sur votre téléphone se recharge d'elle-même, au même endroit :

   ```
   10:14:20  index.html changed. Rebuilt 1 page and reloaded it on 2 open pages.
   ```

   Une feuille de style modifiée s'applique aux pages ouvertes sans rechargement.

Si une page contient une erreur, la plateforme nomme le fichier, la ligne et une correction, et l'aperçu garde la dernière version construite qui fonctionnait :

```
index.html line 5: there is no "site.titel". Did you mean "site.name"?
Nothing changed: the preview still shows the last build that worked.
```

Appuyez sur Ctrl+C pour arrêter la surveillance. Add pages and a shared header ajoute une page et un lien au menu, et Add news, live weather and a form ajoute des nouvelles, la météo en direct et un formulaire.

5. Publier avec ws push

Publier, c'est un commit Git et un `ws push` :

```
$ git add -A && git commit -m "First version"
$ ws push
Built 1 page.
Published version 1 of "Saltmere Marina".

Private link to the sample, open to anyone with it until December 14:
https://saltmere-marina-k7f3q9.sitepeek.page/?key=Vq3x9Lm2Tz8RkP4wNc7bYd

Automatic checks are running. Run ws status for the result.
```

`ws push` affiche aussi des lignes de progression, comme `Building the site.`, omises ici. Chaque publication crée une **version** numérotée : une copie enregistrée de chaque fichier du commit et des réglages du site. `ws versions` les liste, avec le message de chaque commit comme note de la version.

`ws push` ne publie que le travail inclus dans un commit. Il refuse tant qu'un fichier a des changements hors commit, et affiche `Commit your changes first: git add -A && git commit`. Lancez `git status --short` pour voir ce qui n'est pas encore dans un commit. Tant que vous ne faites pas de push, vos commits sont seulement dans votre dossier : faites un push à la fin de la journée si le travail est prêt pour les gens qui ont le lien.

Les vérifications automatiques testent la vitesse des pages, l'accessibilité et les liens, ainsi qu'un envoi d'essai dans chaque formulaire et les traductions manquantes sur les sites qui en ont. Un échantillon est mis en ligne d'abord, et vérifié en arrière-plan, ce qui prend quelques minutes. Lancez `ws status` jusqu'à ce qu'il affiche le résultat :

```
$ ws status
"Saltmere Marina" (saltmere-marina)
This clone includes published version 1.
No unpublished commits.
No uncommitted changes.
Version 1 passed checks: speed, accessibility, links.
```

Pendant les vérifications, sa dernière ligne affiche `Automatic checks for version 1 are running.`

Si une vérification échoue, l'échantillon reste en ligne et le résultat nomme le problème. Corrigez-le et refaites un push.

Si une version plus récente a été publiée et que votre dossier ne l'a pas encore, depuis un autre ordinateur ou le panneau du propriétaire, le push est refusé avec `Not published: someone published first. Run git pull, then ws push.` Publish with Git and work with other builders traite d'un deuxième ordinateur et de deux personnes qui changent le même fichier.

6. Partager le lien de l'échantillon

L'équipe des ventes envoie le lien privé d'un échantillon à l'entreprise.

1. Copiez le lien que `ws push` a affiché. Les publications suivantes gardent le même lien, et `ws site` l'affiche de nouveau.
2. Donnez-le à l'équipe des ventes. Toute personne qui a le lien peut ouvrir l'échantillon : envoyez-le seulement aux personnes à qui il est destiné.

Le lien dure 30 jours après la dernière publication. Quand il prend fin, le lien cesse de fonctionner et l'échantillon invendu est supprimé avec toutes ses versions. Pour garder un échantillon plus longtemps :

```
$ ws site --expires 60d
```

Quand l'entreprise achète son site, le personnel le marque comme acheté dans le panneau des opérations, `ops.websitestrategy.com`, et y rattache le domaine de l'entreprise. Dès lors, `ws push` affiche cette adresse, et les vérifications doivent réussir avant chaque mise en ligne.

7. Répondre à une demande du propriétaire

Un échantillon n'a pas de propriétaire : sur votre premier site, `ws requests` affiche `No open requests for "Saltmere Marina".` Une fois le site acheté, son **propriétaire** le gère dans le panneau du propriétaire, à `app.websitestrategy.com`. Pour demander un changement, le propriétaire ouvre **Change**, pointe une partie d'une page et tape ce qui doit changer. Cela crée une **demande** numérotée, et les créateurs du site reçoivent un courriel.

1. Listez les demandes ouvertes, dans le dossier du site :

```
$ ws requests
Open requests for "Saltmere Marina":
  1  Dana MacLeod, 10:02 AM   "Diesel is $1.89 a litre now."
```

2. Affichez la demande, avec le fichier et la ligne pointés :

```
$ ws requests 1
Request 1 from Dana MacLeod (owner), 10:02 AM, on version 3:
  "Diesel is $1.89 a litre now."

Dana selected this sentence on saltmeremarina.ca/news/fuel-dock:
  "Diesel is $1.79 a litre."
It comes from _data/news/2026-11-02-fuel-dock.md, line 6.
Its French copy: _data/fr/news/2026-11-02-fuel-dock.md, line 5.
Rebuilt from what Dana's browser showed: https://ops.websitestrategy.com/sites/s-
kqm7qrxztcwxrrpdnfvbcebszd/requests/1/picture.png
```

   L'image correspond à ce que le propriétaire a vu.
3. Mettez le dossier à jour. Les changements faits dans le panneau du propriétaire arrivent comme des commits, et `ws push` refuse tant que vous ne les avez pas :

```
$ git pull
```

4. Faites le changement et vérifiez-le avec `ws preview`.
5. Faites un commit, puis un push comme réponse à cette demande :

```
$ git add -A && git commit -m "diesel price"
$ ws push --request 1
Built 10 pages.
Checks passed: speed, accessibility, links, forms, translations.
Published version 4 of "Saltmere Marina" at https://saltmeremarina.ca.
Dana MacLeod gets an email: request 1 is done.
```

Le changement est mis en ligne dès que les vérifications réussissent. `ws undo N` inverse seulement ce que la version N a changé, et `ws push --needs-approval` montre un changement au propriétaire avant les visiteurs. Answer owner requests and ask for approval traite des deux. Les autres parties du guide, en anglais, couvrent les pages, les données, les langues, le travail à plusieurs et l'approbation; la référence liste chaque commande.