

Bâtir une Application de Gestion des Problèmes

Gestion des Problèmes avec l'API ACC

Le guide du développeur pour la mise en place du projet, basé sur NodeJS.

Notre Mission : Maîtriser l'API des Problèmes ACC

Ce guide vous accompagnera pas à pas dans la configuration et la création d'une application web pour interagir avec les **problèmes (Issues)** d'Autodesk Construction Cloud.

Objectifs d'Apprentissage Clés



Obtenir les données générales et complexes des problèmes (titre, attributs personnalisés, commentaires, documents liés, etc.).



Afficher les données dans une visionneuse sous forme de tableau, avec une structure aplatie ("flatten stream") pour les données complexes.



Implémenter des fonctionnalités d'exportation des données vers un fichier CSV.



Permettre l'importation de nouveaux problèmes à partir d'un fichier CSV.

Note | Ce guide se concentre sur la **Partie 1 : Introduction et Configuration de l'Application.**

Anatomie d'un "Issue" dans Autodesk Construction Cloud

Les "Issues" sont une fonctionnalité centrale et générique d'ACC. Au-delà d'un simple ticket, ils sont hautement configurables via l'interface et accessibles via l'API.

Éléments Clés Configurables :

- **Types & Sous-types:** Créez des catégories et des types de problèmes personnalisés.
- **Causes profondes (Root Causes):** Définissez des catégories et des causes spécifiques pour une meilleure analyse.
- **Attributs Personnalisés:** Ajoutez des champs de données sur mesure (texte, listes déroulantes) pour enrichir les problèmes.
- **Permissions:** Gérez les accès par rôle (ex: Admin avec contrôle total, Ingénieur en lecture seule).
- **Modèles (Templates):** Créez des modèles de problèmes pour standardiser la saisie.
- **Statuts:** Suivez le cycle de vie d'un problème à travers des statuts prédéfinis.

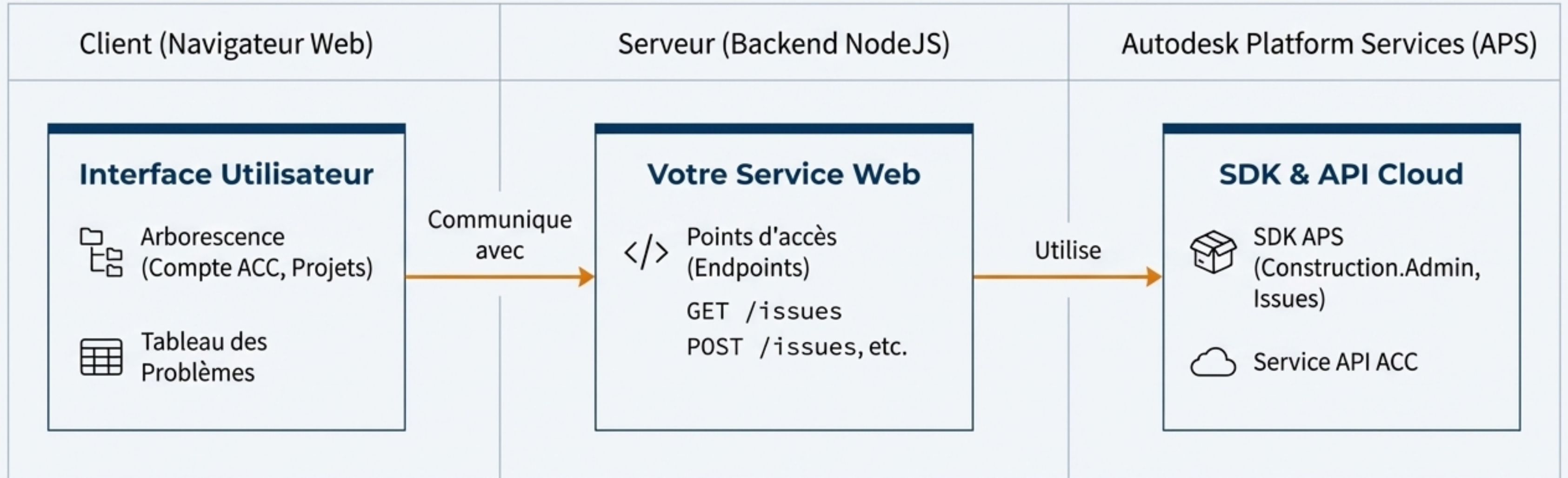
The image displays three screenshots from the Autodesk Construction Cloud (ACC) interface, illustrating the configuration options for Issues.

Types & Sous-types personnalisés: The top screenshot shows the 'Settings' page for 'Issues'. It lists 'Types' (Issue Name, Type, Subcategory, Task, Subcategory) and 'Subtypes' (Issue Name, Custom Fields, Status Type, Personalized Subtype, Subtype Subtype). Each item has a 'Status' toggle switch.

Attributs Personnalisés sur mesure: The middle screenshot shows the 'Create Custom Attribute' dialog. It includes fields for 'Name', 'Type' (Text, Number, Date, Time, Location, Attachment, File), 'Grouped Type' (Text, Dropdown, Radio, Table), and a 'Description' field.

Permissions granulaires par rôle: The bottom screenshot shows the 'Permissions' page. It displays a table with columns for 'Role', 'View', 'Create', 'Edit', 'Delete', and 'Status'. The roles listed are Project Admin, Project Admin, Engineer, Subcontractor, Subcontractor, Subcontractor, and Subcontractor. The permissions are granular, allowing for different levels of access for each role.

L'Architecture de Notre Application



Notre client communique avec notre backend NodeJS. Le backend ne communique pas directement avec l'API REST, mais utilise le SDK APS pour simplifier les appels vers les services cloud d'Autodesk. Le SDK gère la complexité pour nous.

Vos Références Essentielles

Pour approfondir votre connaissance de l'API Issues, ces ressources sont incontournables.



Field Guide

Une introduction détaillée et conceptuelle de l'API Issues. Idéal pour comprendre les fondamentaux.

developer.autodesk.com/en/docs/acc/v1/guides/issues/



How-to Guides

Des tutoriels et workflows ciblés pour des tâches spécifiques (ex: télécharger une pièce jointe, obtenir une référence).

developer.autodesk.com/en/docs/acc/v1/tutorials/



API Reference

La source de vérité. La documentation exhaustive de chaque endpoint, avec les paramètres, les requêtes et les réponses.

developer.autodesk.com/en/docs/acc/v1/reference/http/

Prérequis : Votre Checklist Avant de Coder



Un compte Autodesk Platform Services (APS): Nécessaire pour créer et gérer vos applications.



Une application APS avec ses identifiants: Nous allons la créer ensemble à l'étape suivante.



Un compte Autodesk Construction Cloud (ACC): Pour les tests et le développement.

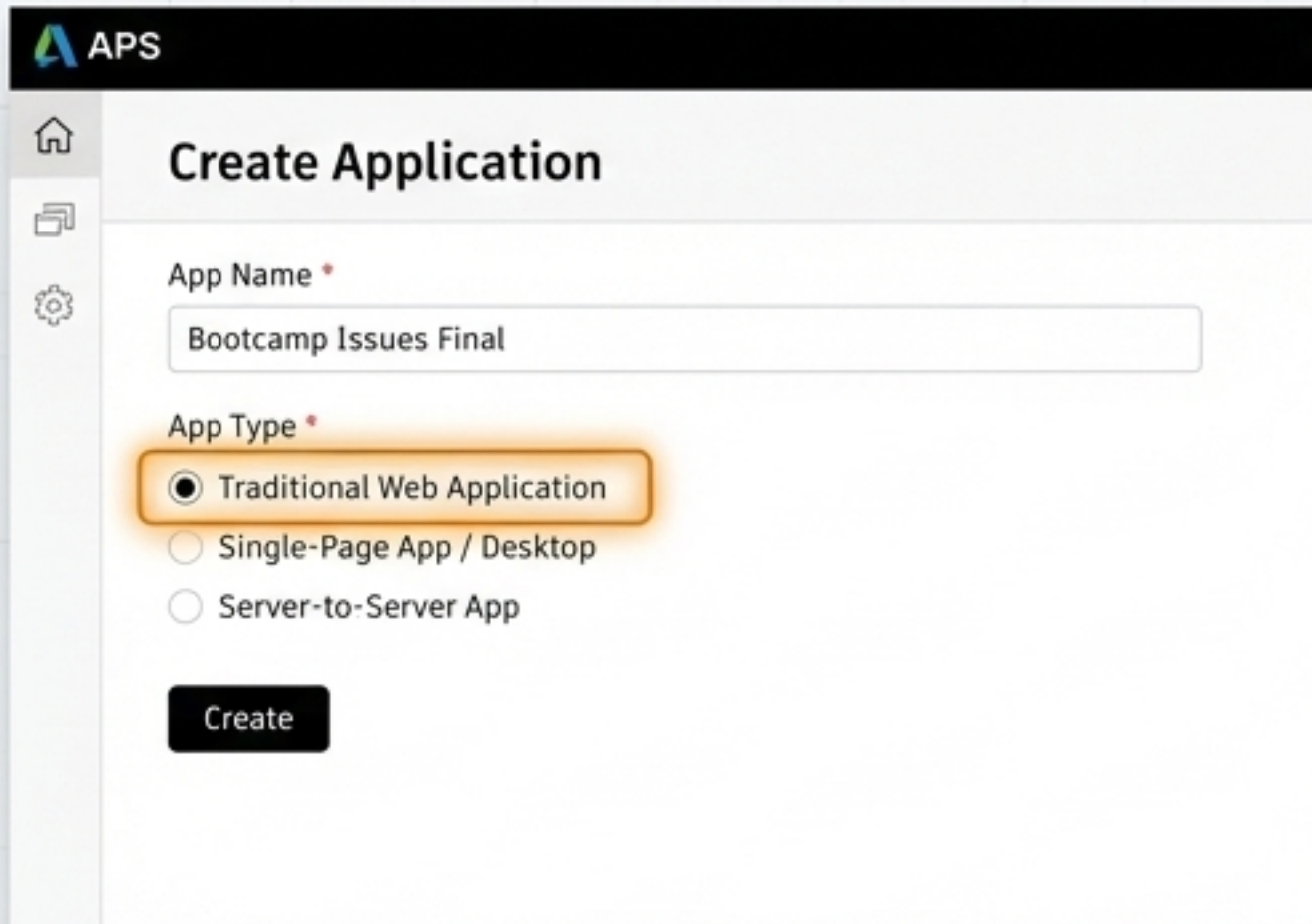


Conseil de Pro

N'utilisez jamais un compte ACC de production pour le développement. Vous pourriez altérer des données réelles. Créez un compte "sandbox" ou de test pour suivre ce tutoriel en toute sécurité.

Étape 1 : Créer votre Application APS

1. Connectez-vous à `aps.autodesk.com`.
2. Naviguez vers 'Mes Applications'.
3. Cliquez sur 'Créer une application'.
4. Remplissez le nom de l'application (ex: 'Bootcamp Issues Final').
5. **Sélectionnez le type d'application : 'Traditional Web Application'.**



Explication des Types

- **Traditional Web App:** Notre choix. Pour des applications avec un backend (serveur) et un frontend (client).
- **Single-Page App / Desktop:** Pour des applications qui s'exécutent entièrement côté client, sans contrôle direct du backend.
- **Server-to-Server App:** Pour des automatisations pures, sans interaction utilisateur ni connexion manuelle.

Résultat: Une fois créée, votre application disposera d'un **Client ID** et d'un **Client Secret**. Gardez-les précieusement.

Étape 2 : Le Provisioning, l'Étape Cruciale

Après avoir créé votre application APS, vous devez autoriser son Client ID à accéder à votre compte ACC. C'est l'étape d'intégration personnalisée, ou 'provisioning'.



Marche à suivre:

1. **Copiez** le **Client ID** de votre application APS.
2. Connectez-vous à votre compte **ACC** en tant qu'administrateur.
3. Allez dans **Account Admin -> Custom Integrations**.
4. Cliquez sur **Add Custom Integration**.
5. **Collez** votre Client ID et donnez un nom à l'intégration.

Avertissement sur les Permissions:

Vous verrez un avertissement indiquant que l'application peut accéder à vos données. C'est normal pour une 'Traditional Web Application'. Cliquez sur "Proceed Anyway". Cette alerte n'apparaît pas pour les applications de type "Server-to-Server".



Étape 3 : Configurer l'Environnement de Développement



Node.js

Rôle:

L'environnement d'exécution de notre backend.

Action:

Téléchargez et installez la dernière version LTS depuis `nodejs.org`.



Visual Studio Code

Rôle:

Notre éditeur de code. Léger, puissant et open-source.

Action:

Téléchargez-le depuis `code.visualstudio.com`.



Postman (Optionnel)

Rôle:

Utile pour tester nos endpoints d'API à la fin du développement.

Action:

Peut être utilisé en ligne ou téléchargé.

Exécution : Initialiser le Projet

Ouvrez VS Code et son terminal intégré. Nous allons maintenant créer le squelette de notre projet NodeJS.

Étape 1 : Créer et ouvrir le dossier du projet

Via l'interface de VS Code ou en ligne de commande, créez un nouveau dossier (ex: bootcamp-issues-project) et ouvrez-le.

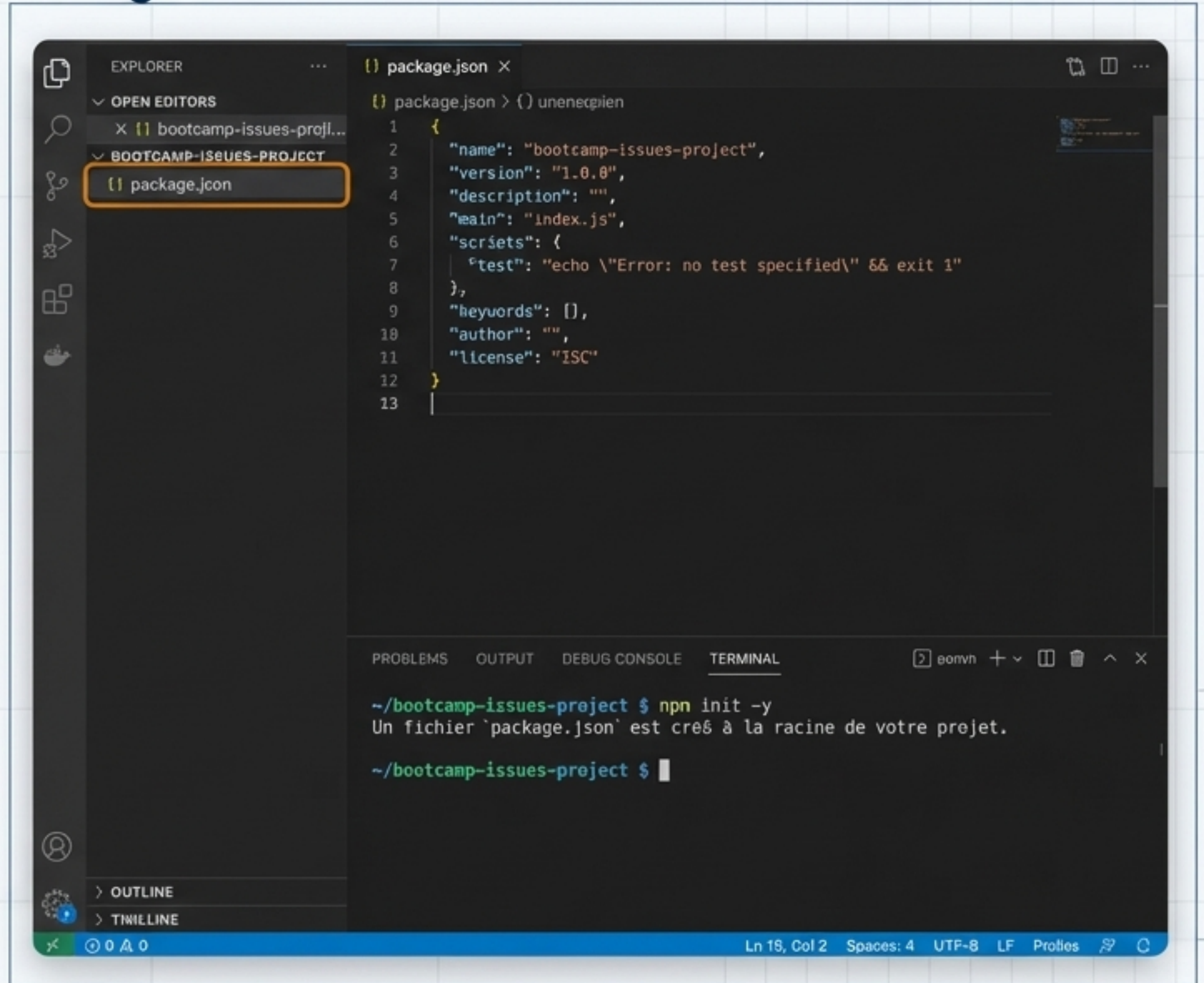
Étape 2 : Initialiser le projet Node.js

Dans le terminal de VS Code, exécutez la commande suivante. Elle crée le fichier `package.json` qui décrit notre projet.

```
npm init -y
```

Résultat Attendu :

Un fichier `package.json` est créé à la racine de votre projet. C'est le cœur de la configuration de notre application NodeJS.



Installation des Dépendances Clés

Nous allons installer deux types de paquets : les bibliothèques pour notre serveur web et les SDKs Autodesk pour communiquer avec l'API.

Partie 1 : Dépendances du Serveur

Ces paquets gèrent le serveur web, les variables d'environnement et les sessions utilisateur.

```
npm install dotenv express cookie-session
```

- **dotenv:** Charge les variables d'environnement (comme nos clés API) depuis un fichier `.env`.
- **express:** Le framework web pour créer notre serveur et nos endpoints.
- **cookie-session:** Gère les sessions basées sur les cookies pour maintenir l'utilisateur connecté.

Partie 2 : SDKs Autodesk Platform Services

Ces paquets officiels simplifient grandement les appels à l'API APS.

```
npm install autodesk-forge-sdk @aps_sdk/authentication @aps_sdk/data-management  
@aps_sdk/construction-admin @aps_sdk/construction-issues
```

Fondation Terminée. Prêts à Construire le Backend.

Récapitulatif de nos acquis

- ✓ Compréhension de l'architecture cible.
- ✓ Création et provisioning de l'application APS.
- ✓ Environnement de développement configuré.
- ✓ Projet NodeJS initialisé avec toutes les dépendances nécessaires.

Prochaines Étapes

(couvertes dans la Partie 2 du tutoriel)

1. **Mettre en place l'authentification** sur le serveur.
2. **Créer les endpoints** pour la navigation (Hubs, Projets).
3. **Développer les endpoints** pour les Problèmes (Issues) et leurs paramètres.
4. **Ajouter les endpoints** pour l'administration du projet (Project Users).

