



Welcome home !

Règlement intérieur

Poster une issue

JavaScript

Backend & Strapi

Prérequis

👋 Semaine 1

Introduction au concept de backend

Bases de Strapi

👋 Semaine 2

Les références

Customisation des routes

🎥 Vidéo

Destructuring

L'arborescence des projets Strapi

Customiser une route du CRUD déjà existante

Entity Service API, strapi, super et ctx

👋 Semaine 3

L'authentification

Les différentes BDD utilisables

Stocker des fichiers

Policy et nouvelles routes

👋 Semaine 4

Recherche et tri

Git

Git & Github

Northflank : héberger le serveur et la BDD

HTML / CSS

Entity Service API, strapi, super et ctx

Dans le cours précédent, nous avons vu un exemple, dans lequel nous étendions le comportement par défaut d'une route : la route `create` d'une collection.

Pour cela nous étions parti du fichier `src > api > reservation > controllers > reservation.js` qui contenait le code suivant :

```
"use strict";

/**
 * reservation controller
 */

const { createCoreController } = require("
module.exports = createCoreController("api
```

Nous avons ajouté un deuxième argument à la fonction `createCoreController` :

```
"use strict";

/**
 * reservation controller
 */

const { createCoreController } = require("
module.exports = createCoreController(
  "api::reservation.reservation",
  ({ strapi }) => ({
    async create(ctx) {}
  })
);
```

Nous avons utilisé les variables `strapi`, `ctx` `super` pour effectuer différentes actions dans notre controller étendu. Voyons plus en détail à quoi elles peuvent servir.