

Zoho Creator 6 - Tutoriel pour comprendre les API personnalisées

Par Zoho

Date de publication : 15 mars 2024

Cette page d'aide s'adresse aux utilisateurs de Creator 6. Trouvez votre **version de Creator**.

Pour réagir au contenu de cet article, un espace de dialogue vous est proposé sur le forum. **Commentez**

I - Que couvre cette page ?.....	3
II - Disponibilité.....	3
III - API personnalisées dans Creator.....	3
III-A - Fonctionnalités des API personnalisées dans Creator.....	4
III-B - Comment les API personnalisées sont-elles construites dans Creator ?.....	4
III-C - Composants d'une API personnalisée dans Creator.....	4
III-C-1 - Entrées de base.....	4
III-C-2 - URL d'extrémité.....	5
III-C-2-a - Fonctionnalités de l'URL d'extrémité.....	5
III-C-3 - Requête.....	6
III-C-3-a - Méthodes de requête pour l'API personnalisée.....	6
III-C-3-b - Authentification.....	6
III-C-3-c - Portée de l'utilisateur.....	7
III-C-3-d - Type de contenu.....	7
III-D - Réponse.....	8
III-E - Fonction.....	10
III-F - Résumé.....	11
IV - Logs API.....	11
V - Champs d'application pour invoquer l'API personnalisée.....	12
VI - Voir comment cela fonctionne.....	12
VII - Cas d'utilisation.....	12
VIII - Guide de navigation pour la création d'API personnalisées.....	13
IX - Points à noter.....	13
X - Limitations.....	13
XI - Sujets connexes.....	13
XII - Remerciements Developpez.com.....	14

I - Que couvre cette page ?

Découvrez les API personnalisées et comment définir leurs actions à l'aide des fonctions Deluge, qui peuvent ensuite être exposées en tant qu'API et appelées à partir d'une plateforme externe. Cela permettra de maximiser l'efficacité opérationnelle entre plusieurs plateformes, y compris Zoho Creator, et de réduire la difficulté de programmer des opérations complexes à travers des systèmes externes.

II - Disponibilité

API personnalisées :


- Ne peuvent être créées que dans les **plans payants** de Creator, alors que les utilisateurs des **plans d'essai** peuvent les invoquer.
- Ne peuvent être créées, autorisées et gérées que par le **super administrateur et les administrateurs**, tandis que les autres **utilisateurs** peuvent y accéder.
- Sont disponibles dans les **centres de données** États-Unis (.com), de l'Europe (.eu), de l'Inde (.in) et de l'Australie (.au).

III - API personnalisées dans Creator

Dans Creator, les API personnalisées permettent aux développeurs d'obtenir des fonctionnalités spécifiques adaptées à leurs besoins en utilisant les **fonctions** existantes dans leurs applications. Alors que les API de **développeur** (API REST) exécutent une action fixe (prédéfinie) lorsqu'elles sont invoquées, les API personnalisées peuvent être scriptées pour exécuter des actions définies par l'utilisateur à partir de plates-formes externes. Cela signifie que les utilisateurs peuvent effectuer des actions autres que celles prédéfinies sur les données de leur application Creator à partir d'applications tierces en invoquant leurs fonctions en tant qu'API personnalisée. Cette personnalisation garantit que l'API personnalisée s'aligne précisément sur les besoins de votre organisation, optimisant ainsi les performances et l'efficacité.

Les API personnalisées vous permettent également d'intégrer vos applications à des services tiers, ce qui vous permet d'étendre les capacités de votre application en tirant parti de ressources externes. En outre, ces API encapsulent votre logique d'entreprise, ce qui facilite sa maintenance et sa mise à jour en un seul endroit. Cela signifie que toute modification des processus commerciaux peut être mise en œuvre dans l'API elle-même, sans nécessiter de modifications de l'ensemble de l'application.

Dans Creator, l'API personnalisée peut être créée et gérée dans le *Custom API builder*. Ce constructeur vous permet de définir les actions à exposer en tant qu'API à l'aide d'une **fonction Deluge** existante (qui a l'action définie) et d'invoquer l'API à partir d'une plateforme externe en utilisant l'URL de point final générée automatiquement à l'adresse .

 Seuls le super administrateur et les administrateurs peuvent accéder au *constructeur d'API* personnalisées pour créer et gérer des API personnalisées.

Après avoir invoqué votre API personnalisée, vous pouvez consulter le statut de vos **hits API (succès/échec)** et leurs journaux d'erreurs dans la page *Résumé de l'API* .

Remarque :

- Le nombre d'accès à l'API sera comptabilisé séparément des accès à l'API REST en termes de **tarification**.
- Actuellement, vous ne pouvez utiliser que les **fonctions Deluge** déjà configurées en tant qu'API personnalisées. Cela signifie que vous ne pouvez pas invoquer des **fonctions cloud** écrites à l'aide de scripts **Java** et **NodeJS** en tant qu'API personnalisée.

- Vous ne pouvez pas tester vos API personnalisées dans un **environnement de développement** puisque seules les fonctions Deluge disponibles dans l'**environnement de production** peuvent être exposées en tant qu'API personnalisées.

III-A - Fonctionnalités des API personnalisées dans Creator

- Les API personnalisées peuvent permettre d'invoquer des opérations qui ne sont autrement possibles que dans Creator. Par exemple, l'envoi de **notifications push**, la récupération de résultats à partir de **champs de formulaire Alet** l'exécution d'autres actions liées aux données, etc.
- Les utilisateurs peuvent invoquer les **fonctions Deluge** de Creator dans votre API personnalisée au lieu de rescrire une fonction Deluge ou de créer une autre API personnalisée.
- Créer des API personnalisées sécurisées en mettant en œuvre un **mécanisme d'authentification** pour s'assurer que l'accès aux données est contrôlé et sécurisé, réduisant ainsi le risque d'accès non autorisé ou de violation des données.
- Exécuter **plusieurs** opérations logiques par le biais d'une seule fonction.
- Analyser et gérer les **journaux des API personnalisées** comme une seule opération en un seul endroit lors de l'exécution d'une opération de grande envergure par le biais de plusieurs appels d'API ou de scripts externes.

III-B - Comment les API personnalisées sont-elles construites dans Creator ?

- **Constructeur d'API personnalisées** : une interface sécurisée et facile à construire qui permet de créer une **API personnalisée prête à l'emploi** qui facilite l'échange de données entre **Creator** et les systèmes externes.
- **Fonctions Deluge** : vos **fonctions Deluge** existantes peuvent être invoquées dans les API personnalisées.

III-C - Composants d'une API personnalisée dans Creator

III-C-1 - Entrées de base

The screenshot shows the 'Custom API Builder' interface. On the left, a sidebar contains a list of tabs: 'Basic Details' (selected), 'Request', 'Response', 'Actions', and 'Summary'. The main area is titled 'Basic Details' and contains the following fields:

- Display Name:** A text input field containing 'Before Trip OAuth'.
- Link Name:** A text input field containing 'before-trip'.
- Description (optional):** A text area containing the text: 'Get address data from Teachers, Admin and Students, Categorize area-wise and fetch geoCodes from Zoho Maps. Formatted data sent to GPS Module.'

On the right side, there is a 'Hints' panel with the following text:

- Ensure to name your custom API with terms that are descriptive, memorable, and reflective of their functionality.
- The link name will be appended to the endpoint URL generated.
- The endpoint URL serves as the point of access while sending or receiving information between your Creator app and external services.

L'onglet *Basic Inputs* vous permet de définir votre API personnalisée. Vous pouvez spécifier un **nom** pour votre API personnalisée, ajouter un **nom de lien** qui sera utilisé dans l'URL du point final et décrire l'**objectif** de votre API personnalisée.

Lorsque vous donnez un nom à votre API personnalisée, il est essentiel de choisir des noms descriptifs, mémorisables et qui reflètent leur fonctionnalité.

III-C-2 - URL d'extrémité

L'**URL du point de terminaison de l'API** fait référence à une URL spécifique que l'API personnalisée expose pour permettre la communication avec des systèmes externes. Cette URL définit le chemin que les utilisateurs peuvent utiliser pour accéder à des fonctionnalités particulières fournies par l'API. Dans Creator, l'**URL du point de terminaison de l'API personnalisée** sera automatiquement générée sur la base du **nom du lien API** spécifié dans le format ci-dessous. Cette URL vous permet, ainsi qu'à vos utilisateurs, d'identifier l'objectif spécifique de votre API. Vous pouvez copier et utiliser l'URL à partir des endroits mentionnés dans la section suivante, selon vos besoins.

`https://www.zohoapis.com/creator/custom/<appadmin_name>/<customAPIname>`

III-C-2-a - Fonctionnalités de l'URL d'extrémité

Comment obtenir l'URL du point final ?

Vous pouvez copier l'URL du point de terminaison à partir des endroits suivants :

- Onglet Résumé
- Carte API personnalisée respective
- Vue détaillée

La vidéo suivante montre les endroits à partir desquels vous pouvez copier l'URL du point de terminaison.

<https://workdrive.zohoexternal.com/embed/bx83g5f610175a9904abc934ae9f39d79f486?toolbar=false&start=0>

Qui peut accéder à l'URL du point final ?

L'accès est basé sur l'étendue de l'utilisateur définie dans l'onglet **Demande**. Si vous avez choisi l'option **Admin Only**, seuls le **super administrateur** et les **administrateurs de** l'application peuvent accéder à l'API personnalisée et l'invoquer. Si vous avez choisi l'option **Utilisateurs sélectifs**, seuls les utilisateurs spécifiés (ainsi que le super administrateur et les administrateurs) peuvent accéder à l'API personnalisée et l'invoquer.

Remarque: si un utilisateur non autorisé tente d'invoquer l'API personnalisée à l'aide de l'URL du point de terminaison, une **réponse du système** (c'est-à-dire la réponse du créateur) s'affichera.

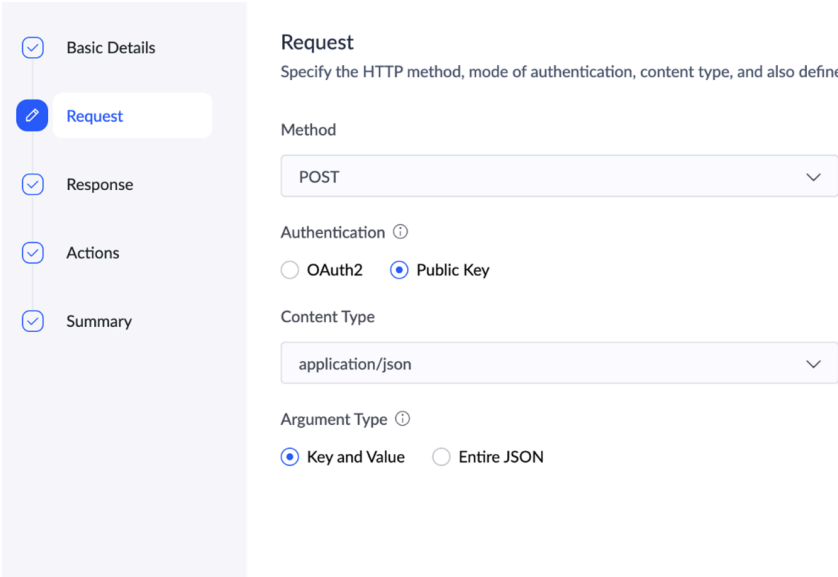
Où l'URL du point final peut-elle être invoquée ?

Vous pouvez l'utiliser dans n'importe quel service tiers pour invoquer l'API personnalisée.

Remarque:

- L'URL du point de terminaison doit être copiée et collée dans la plateforme externe requise pour invoquer l'API personnalisée.
- Lorsque vous modifiez le nom du lien de votre API personnalisée, son champ d'application ou sa méthode d'authentification, l'aperçu de l'URL d'extrémité sera automatiquement mis à jour dans le *Custom API builder*. Toutefois, vous devez mettre à jour manuellement l'URL du point de terminaison lorsqu'elle a été invoquée. Si vous invoquez l'API personnalisée sans mettre à jour l'URL d'extrémité, une réponse du système (erreur) s'affichera.

III-C-3 - Requête



Request
Specify the HTTP method, mode of authentication, content type, and also define the API scope.

Method
POST

Authentication ⓘ
 OAuth2 Public Key

Content Type
application/json

Argument Type ⓘ
 Key and Value Entire JSON

Dans l'onglet *Requête*, vous pouvez spécifier la méthode, l'authentification et le type de contenu utilisés dans le corps de la demande.

- [Méthodes de requête](#)
- [Authentification](#)
- [Portée de l'utilisateur](#)
- [Type de contenu](#)

III-C-3-a - Méthodes de requête pour l'API personnalisée

Ces méthodes vous permettent de définir le type d'action que vous souhaitez effectuer lors du déclenchement d'une demande d'API. Le *générateur d'API personnalisée* vous permet de choisir parmi quatre méthodes : **GET**, **POST**, **PUT** et **DELETE**.

- **GET**: Cette méthode vous permet de demander et d'extraire des données de l'URL d'extrémité spécifiée. Dans une requête GET, les données sont généralement envoyées sous la forme de paramètres de requête ajoutés à la fin de l'URL.
- **POST**: Cette méthode vous permet de soumettre des données à traiter à l'URL de destination spécifiée. Dans une requête POST, les données sont généralement envoyées dans le corps de la requête, souvent dans des formats tels que JSON ou des données de formulaire. Ici, les données ne sont pas ajoutées à l'URL.
- **PUT**: Cette méthode est utilisée pour mettre à jour ou remplacer une ressource existante ou pour créer une nouvelle ressource à l'URL spécifiée.
- **DELETE**: Cette méthode est utilisée pour retirer ou supprimer une ressource spécifique du serveur à une URL donnée.

III-C-3-b - Authentification

Dans Creator, vous pouvez sélectionner le **type d'authentification**: **OAuth2** ou **Clé publique**. Par défaut, OAuth, qui signifie "**Open Authorization**", permet d'autoriser et d'authentifier les appels d'API. Il permet à vos applications Creator d'accéder en toute sécurité à des ressources externes, ce qui évite de demander un nom d'utilisateur et un mot de passe à chaque fois qu'un utilisateur se connecte. Si cette option est choisie, seuls les utilisateurs spécifiés dans l'étendue de l'utilisateur peuvent accéder à l'API personnalisée et l'invoquer.

Si vous choisissez l'option **Clé publique**, tout le monde peut accéder à l'API personnalisée et l'invoquer.

III-C-3-c - Portée de l'utilisateur

En tant que **superadministrateur** ou **administrateur**, vous pouvez définir le niveau de contrôle d'accès accordé à vos administrateurs et utilisateurs sélectifs à l'aide de l'*étendue de l'utilisateur*, c'est-à-dire que vous pouvez autoriser des utilisateurs spécifiques à accéder à l'URL du point de terminaison et à invoquer l'API. Vous pouvez choisir d'accorder le contrôle d'accès aux **seuls administrateurs de** vos applications Creator ou à **certains utilisateurs** en spécifiant leurs adresses électroniques respectives.

III-C-3-d - Type de contenu

Remarque : vous ne pouvez sélectionner le **type de contenu** que pour les méthodes **POST** et **PUT**.

Le type de contenu est utilisé pour spécifier le format et la structure des données envoyées ou reçues dans une demande ou une réponse à destination et en provenance de vos applications Creator et des services externes. Cela aide le service externe à comprendre comment la charge utile de la demande doit être interprétée. Creator vous permet de sélectionner un type de contenu parmi deux formats :

- **Application/JSON**: ce type de contenu peut être choisi pour indiquer que les données de la demande API sont au format **JSON(JavaScript Object Notation)**. Cela indique au service externe que les données contenues dans le corps de la demande sont au format JSON et qu'elles doivent être traitées en conséquence.

JSON est un format d'échange de données largement utilisé pour représenter des objets de données, des tableaux et des paires clé-valeur, et il est à la fois lisible par l'homme et par la machine. Il est couramment utilisé pour transmettre des données structurées entre un client et un serveur, ce qui en fait un choix idéal pour la communication API.

- **Multipart/form-data**: Ce type de contenu est utilisé dans les demandes d'API pour envoyer des données binaires ou textuelles, telles que des fichiers, à un service externe. Dans une requête **multipart/form-data**, les données sont divisées en plusieurs parties, chaque partie ayant son propre type de contenu et ses propres en-têtes. Cela permet d'inclure non seulement les données elles-mêmes, mais aussi des informations supplémentaires telles que les noms de fichiers, les noms de champs et les types de contenu pour chaque partie. Une chaîne de délimitation est utilisée pour séparer les parties de la demande. Cette chaîne est unique et est spécifiée dans les en-têtes de la demande pour indiquer où commence et où finit chaque partie.

Le format multipart/form-data est couramment utilisé lors de la création de points d'extrémité d'API permettant aux utilisateurs de télécharger des fichiers ou lors de l'utilisation de formulaires comprenant des champs de téléchargement de fichiers. Lorsqu'un utilisateur soumet un formulaire contenant un champ de saisie de fichier, l'API personnalisée génère une requête "multipart/form-data" pour envoyer les données au serveur externe.

Remarque :

- Les données multipart/form-data ne peuvent pas avoir de données imbriquées comme JSON, ce qui signifie qu'elles contiennent simplement des paires "clé-valeur" séparées.
- Il peut y avoir plusieurs entrées sur une même clé, contrairement à JSON.

Si le type de contenu **Application/JSON** est choisi, vous pouvez sélectionner le type d'argument comme l'une des deux options suivantes.

- **Clé et valeur** : Ce type vous permet de transmettre des paramètres d'entrée à votre point final d'API sous la forme d'une paire clé-valeur, et peut être utilisé lorsque vous devez transmettre un petit nombre de paramètres distincts et nommés dans votre demande d'API.

Sample Request :

```
1. {
```

```

2.   "name": "John",
3.   "age": 30,
4.   "email": "john@zylker.com"
5. }

```

Dans l'exemple ci-dessus :

- "name", "age" et "email" sont des clés.
- "John", 30 et " john@zylker.com " sont les valeurs correspondantes.
- **Entire JSON**: ce type vous permet de transmettre un objet JSON complet et non structuré, y compris toutes les paires clé-valeur nécessaires, en tant qu'argument unique. Vous pouvez utiliser cette approche lorsque vous traitez des appels de type "webhook".

Exemple de demande:

```

1.   {
2.     "user": {
3.       "id": 12345,
4.       "name": "Alisa",
5.       "email": "alisa@zylker.com",
6.       "age": 28,
7.       "address": {
8.         "street": "123 Main St",
9.         "city": "Anytown",
10.        "country": "Zylkerland"
11.      },
12.      "orders": [
13.        {
14.          "order_id": 1001,
15.          "total_amount": 75.5,
16.          "status": "delivered"
17.        },
18.        {
19.          "order_id": 1002,
20.          "total_amount": 102.2,
21.          "status": "pending"
22.        }
23.      ]
24.    },
25.    "status": "success",
26.    "message": "Données de l'utilisateur récupérées avec succès"
27. }

```

Dans l'exemple ci-dessus :

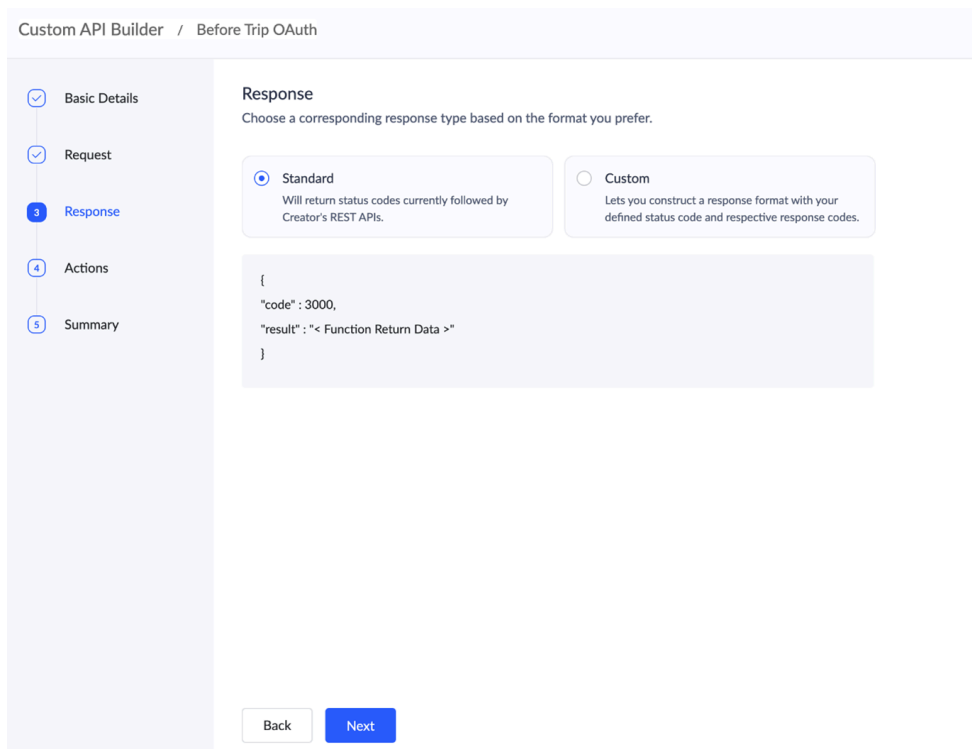
- La **réponse JSON** contient plusieurs éléments imbriqués.
- **"user"** contient des informations sur l'utilisateur, notamment son identifiant, son nom, son adresse électronique, son âge, son adresse et une liste de commandes.
- **"address"** est un objet imbriqué dans les détails de l'utilisateur, contenant des informations relatives à l'adresse.
- **"orders"** est un tableau contenant plusieurs objets représentant les différentes commandes passées par l'utilisateur, chacune avec ses propres attributs tels que l'identifiant de la commande, le montant total et l'état.
- **"status"** et **"message"** fournissent des informations supplémentaires sur l'état général de la demande/ réponse de l'API.

Lorsque vous choisissez **Entire JSON** pour cette API personnalisée, veillez à ne spécifier qu'un seul argument de type chaîne de données ici chaque clé est associée à une valeur spécifique.

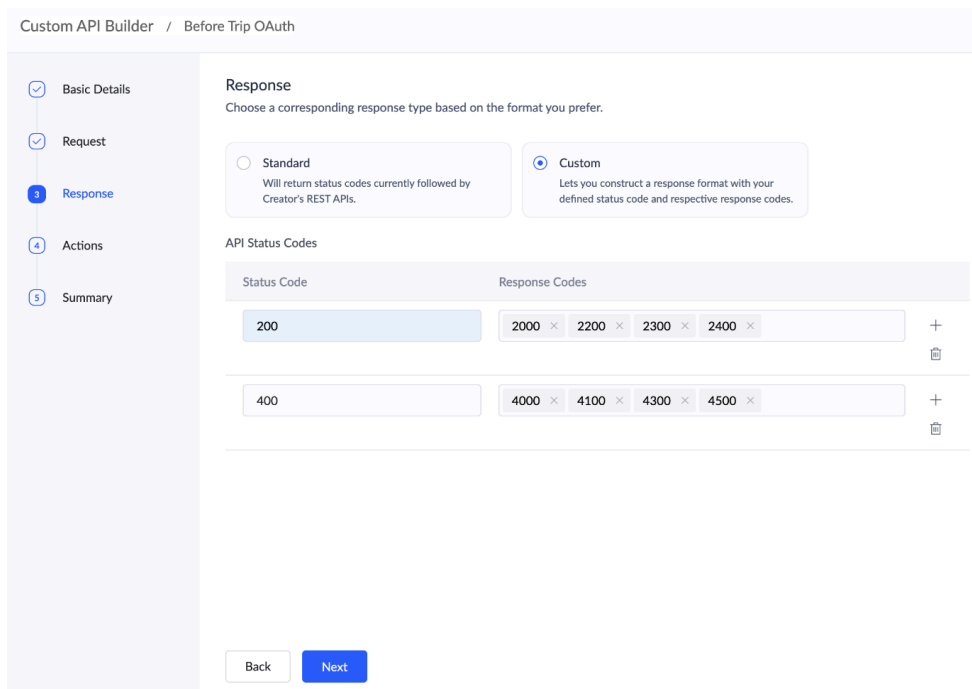
III-D - Réponse

Dans cet onglet, vous pouvez choisir le type de réponse correspondant : Réponse **standard** ou Réponse **personnalisée**.

- **Réponse standard:** Cette réponse renvoie les codes d'état standard actuellement suivis par les **API REST** de Creator. Lorsque cette option est choisie, un exemple de réponse s'affiche ci-dessous.



Réponse personnalisée: Cette réponse vous permet de définir et de spécifier un **code d'état** personnalisé et les **codes de réponse** correspondants en fonction de vos besoins.



- Vous pouvez spécifier jusqu'à **50 codes de réponse** pour chaque **code d'état**.
- Lorsque vous spécifiez des codes personnalisés, le type de retour de la **fonction Deluge** à invoquer doit être **Map**.
- Vous devez associer les **codes de réponse** spécifiés à leurs **messages descriptifs** dans la fonction Deluge.

III-E - Fonction

Custom API Builder / Before Trip OAuth

- Basic Details
- Request
- Response
- 4 Actions**
- 5 Summary

Actions

Specify the details of the function to be executed when the custom API is invoked.

Application

Namespace

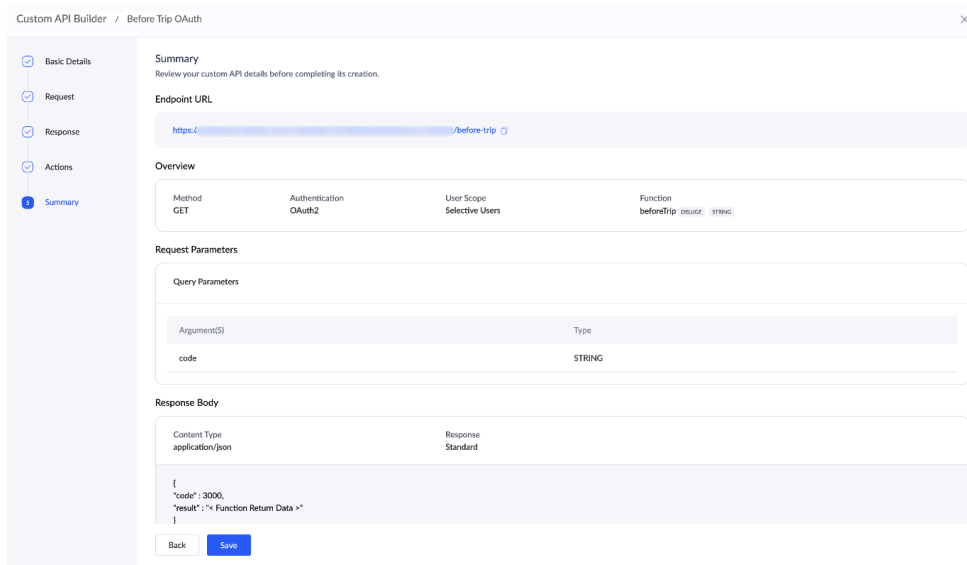
Function Custom API supports only Deluge functions.

DELUGE
STRING

Dans l'onglet *Actions*, vous pouvez sélectionner l'**application Creator** et l'**espace de noms** qui contient la **fonction Deluge** à exécuter lorsque l'API personnalisée est invoquée.

- Lorsque vous créez une **fonction Deluge**, il est recommandé de spécifier un espace de noms et une fonction appropriés. Il sera ainsi plus facile de définir les actions de votre API personnalisée.
- Il est recommandé de faire preuve de prudence lorsque vous incluez des opérations sensibles dans vos fonctions Deluge et que vous donnez finalement accès à des utilisateurs spécifiques pour invoquer votre API personnalisée.

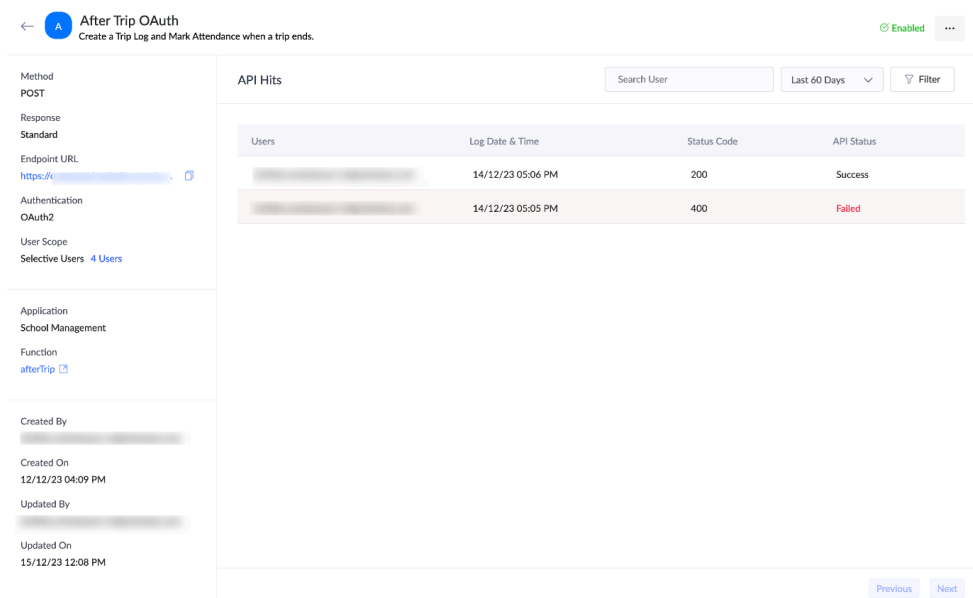
III-F - Résumé



Cet onglet vous permet de revoir les détails de votre API personnalisée avant de terminer sa création. Après avoir ajouté les détails requis dans les onglets précédents, vous pouvez examiner les détails de l'API affichés dans les sections *Overview (Présentation)*, *Request Parameters (Paramètres de la requête)* et *Response Body (Corps de la réponse)*. Si vous devez apporter des modifications, vous pouvez revenir en arrière et le faire. Sinon, vous pouvez enregistrer l'API personnalisée.

Lorsque vous créez une API personnalisée et que vous quittez la page avant de l'enregistrer, l'API personnalisée est enregistrée en tant que **brouillon**.

IV - Logs API



Après avoir créé votre API personnalisée, vous pouvez afficher les détails essentiels de l'API personnalisée, l'**application** dans laquelle elle est utilisée, ainsi que les détails de sa **création** et de sa **mise à jour**. Vous pouvez également consulter le nombre d'**accès** à l'API par les utilisateurs qui l'ont invoquée, la date et l'heure du journal, le

code d'état et l'état de l'API(**succès** ou **échec**). En outre, vous pouvez également consulter les journaux de l'API à différentes périodes : **quotidienne**, **hebdomadaire** et **mensuelle**.

- Les journaux de l'API affichent par défaut les journaux des **30 derniers jours** et au maximum les journaux des **60 derniers jours**.
- Seuls le superadministrateur et les administrateurs peuvent voir les hits de l'API, c'est-à-dire que les utilisateurs ne peuvent pas voir les hits de l'API.

V - Champs d'application pour invoquer l'API personnalisée

Method	Scope
GET, POST, PUT, DELETE	Zohocreator.customapi.EXECUTE

VI - Voir comment cela fonctionne

Supposons que vous ayez créé une application de *gestion scolaire* comportant des modules distincts pour les élèves, les enseignants, les parents et le personnel non enseignant. En tant qu'administrateur de l'application, vous devez traiter les données provenant du dispositif IoT de suivi des bus installé dans vos bus scolaires et accessible aux chauffeurs de bus de votre école. Supposons que vous ayez les deux exigences suivantes.

- Récupérer les adresses de ramassage du personnel, des élèves et du personnel non enseignant dans toutes les zones couvertes avant le début d'une bande.
- Soumettre l'assiduité des élèves et la vitesse moyenne du trajet à la fin de chaque trajet.

Pour réaliser ce qui précède, vous aurez généralement besoin d'invoquer au moins **cinq API**.

- **Trois API Get Records** pour récupérer les adresses dans trois rapports distincts.
- **Deux API Add Records** pour ajouter des détails sur le trajet et marquer les présences de ce jour en tant qu'enregistrements.

En outre, vous devez récupérer les enregistrements des élèves absents et informer les parents respectifs par le biais de notifications SMS.

Pour réduire le nombre d'API créées et obtenir la même fonctionnalité, vous pouvez **créer une API personnalisée** et utiliser une fonction Deluge existante qui a le login combiné à invoquer lorsque l'API personnalisée est appelée.

La vidéo ci-dessous montre l'exécution d'une API personnalisée pour récupérer les adresses de trois rapports distincts.

<https://workdrive.zohoexternal.com/embed/e5jyw328ac5df76d14485b5d4553ce1f88179?toolbar=false&start=0>

VII - Cas d'utilisation

- **Gestion d'un hôpital** : Supposons que vous soyez propriétaire d'une clinique et que vous ayez créé une application de *gestion hospitalière* à l'aide de Creator. Vous souhaitez récupérer des informations vitales à partir de dispositifs médicaux (équipés de capteurs) portés par vos patients, afin que les médecins soient immédiatement informés, surveillent leurs signes vitaux et fournissent une assistance médicale si nécessaire. Vous pouvez configurer une API personnalisée qui est invoquée chaque fois qu'une lecture anormale se produit (en tenant compte de diverses données vitales telles que la fréquence cardiaque, la pression artérielle, les niveaux d'oxygène, et ainsi de suite). L'appareil envoie ces détails via une requête API sous forme de **notification mobile** au médecin concerné en fonction de sa disponibilité. Le médecin assigné peut être invité à remplir un formulaire contextuel dans Creator (pour les petites urgences) afin d'envoyer des

conseils médicaux au personnel médical traitant. Le médecin peut également choisir si un rendez-vous est nécessaire immédiatement et réserver le plus tôt possible. Le patient reçoit l'avis médical prescrit ainsi que les détails du rendez-vous (le cas échéant), via la réponse de l'API. Ces données peuvent également être ajoutées aux formulaires de *prise de rendez-vous et de détails sur le patient*, respectivement.

- **Gestion des approbations:** Supposons que vous ayez créé une application de *gestion de projet* à l'aide de Creator. Vos employés créent des documents de proposition dans un service externe, qui doivent être approuvés par vous (administrateur). En outre, vous devez ajouter un enregistrement dans votre application Creator chaque fois qu'un document est approuvé et conserver les enregistrements des documents rejetés. Vous pouvez créer une API personnalisée qui est invoquée lorsqu'un document est soumis pour approbation à partir du service externe. Cela déclenchera le **flux de travail d'approbation** dans Creator, qui lancera le processus d'approbation des documents et changera le statut du document en "Approuvé". Un enregistrement sera ajouté dans les rapports *Détails du projet* et *Détails du devis* respectivement. Si un document est rejeté, le statut du document sera marqué comme "rejeté" et un enregistrement sera ensuite ajouté dans les rapports *Détails du projet* et *Feedback du document* respectivement.

VIII - Guide de navigation pour la création d'API personnalisées

Sur la **page d'accueil** de votre compte Creator, accédez à la section **Microservices** et sélectionnez l'onglet *Custom API*. Ensuite, cliquez sur **+ Créer une API personnalisée**. Le *constructeur d'API* personnalisées s'affiche. Découvrez comment vous pouvez **créer des API personnalisées** et utiliser leurs fonctionnalités au profit de votre entreprise.

<https://workdrive.zohoexternal.com/embed/bx83gccc1d0965ccf49dd83c53fca33eaea82?toolbar=false&start=0>

IX - Points à noter

- Le nom du lien de votre API personnalisée ne peut pas contenir de caractères spéciaux tels que **!@#\$%^&*.>**.
- Les codes d'état de la réponse personnalisée peuvent être compris entre **100 et 599**.
- Les codes de réponse dans la réponse personnalisée doivent être compris entre **1 et 9999** et ne peuvent pas contenir de caractères spéciaux tels que **!@#\$%^&*.>**.
- Lorsque vous modifiez l'étendue de l'utilisateur de votre API personnalisée en la faisant passer d'**Utilisateurs sélectifs** à **Admins Only**, tous les utilisateurs que vous avez précédemment ajoutés seront supprimés.
- Si vous fermez le *Custom API builder* sans enregistrer, vos modifications seront toujours enregistrées en tant que **brouillon** et vous pourrez continuer à les modifier à tout moment à partir de *Microservices > Custom API > carte d'API personnalisée requise*.
- Lorsque vous invoquez une API désactivée, une réponse vide sera renvoyée, c'est-à-dire que l'invocation d'une API désactivée produira un enregistrement dans les journaux d'API mais ne sera pas enregistrée en tant qu'API hit.
- Avant de pouvoir supprimer une fonction utilisée dans votre API personnalisée, vous devrez supprimer l'API personnalisée associée avant de supprimer la fonction correspondante.

X - Limitations

- Dans l'onglet *Response*, vous pouvez ajouter jusqu'à **50** codes d'état.
- Vos codes de réponse personnalisés ne peuvent contenir que **4 chiffres** au maximum.

XI - Sujets connexes

- **Créer et gérer des API personnalisées**
- **API Rest dans Creator**

XII - Remerciements Developpez.com

Nous tenons à remercier **Malick** pour la mise au gabarit et **TODO** pour la relecture orthographique.