



BY-SYSTEMS

**ECAM**  
BRUSSELS ENGINEERING SCHOOL

# Présentation **WEEK 3**

Van Olffen Victor  
Maillard Corentin  
Mettioui Mourad

# Notre équipe

N'hésitez pas à nous contacter !



Van Olffen Victor  
20253@ecam.be



Maillard Corentin  
21306@ecam.be



Mettioui Mourad  
195019@ecam.be

# Programme

- Extensions
- Packages
- WebHook
- HTTP requests

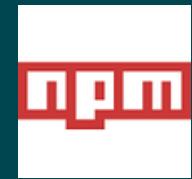
Avec Jira et VisualStudioCode

# Programme

- Extensions
- Packages
- WebHook
- HTTP requests

Avec Jira et VisualStudioCode

# Extensions



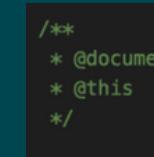
npm Intellisense



Prettier ESLint



Paste JSON as Code



Documents This

# Programme

- Extensions
- Packages
- WebHook
- HTTP requests

Avec Jira et VisualStudioCode

# Packages



TypeScript



Prettier



ESLint

Standard-version

# TypeScript



## Définition

- C'est un langage de programmation open source orienté objet.
- Reprend javascript et contient plus de fonctions comme le typage, les interfaces, ...



## Utilité dans notre projet

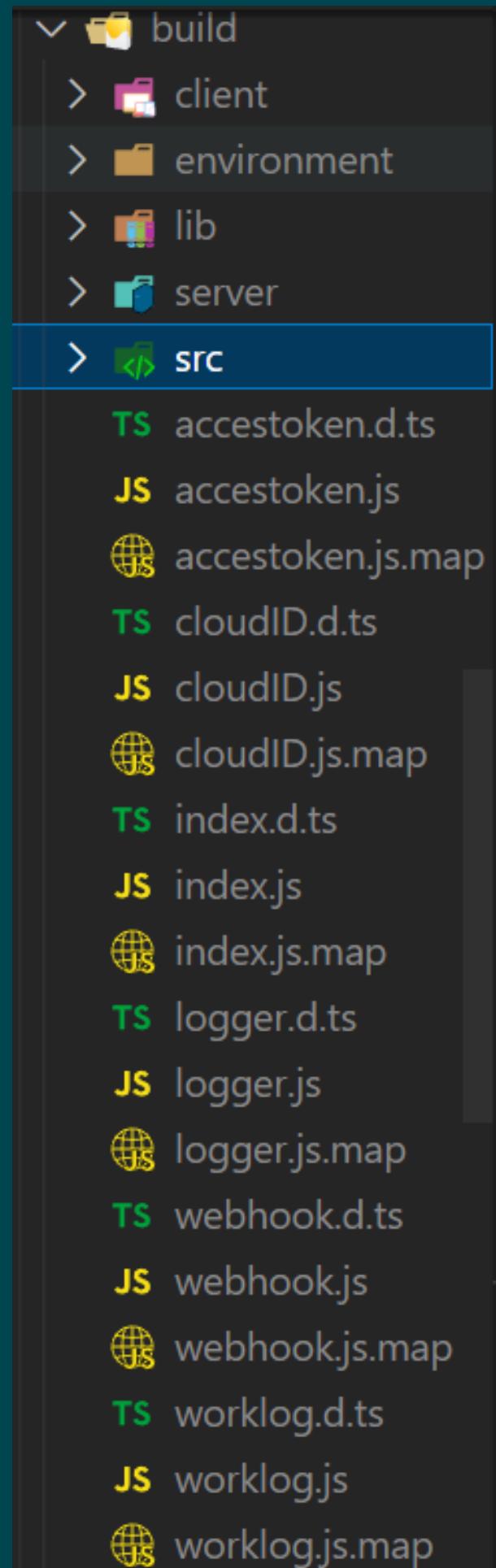
- Crédit à la création de script pour faire du Webhook, des requêtes HTTP, créer des data modèles d'interface.

# TypeScript

## tsconfig.json

```
"compilerOptions": {  
    "lib": ["es2022"],  
    "module": "commonjs",  
    "target": "es2022",  
  
    "strict": true,  
    "sourceMap": true,  
    "declaration": true,  
    "declarationMap": false,  
    "esModuleInterop": true,  
    "resolveJsonModule": true,  
    "skipLibCheck": true,  
    "forceConsistentCasingInFileNames": true,  
    "moduleResolution": "node",  
    "outDir": "./build"  
}
```

commande : tsc -p .



# ESLint



## Utilité dans notre projet

- Analyse notre code pour trouver rapidement les problèmes et les résoudre (pour javascript et typescript).
- Renseigner le project (ici tsconfig.json)

# Prettier



## Utilité dans notre projet

- Permet de formatter notre code quand il est sauvegardé.

```
.prettierrc.json  ×

ts_prj_template > .prettierrc.json > ...

You, 36 seconds ago | 1 author (You)

1  {
2    "arrowParens": "avoid",
3    "bracketSpacing": true,
4    "endOfLine": "lf",
5    "bracketSameLine": false,
6    "jsxSingleQuote": false,
7    "printWidth": 80,
8    "proseWrap": "preserve",
9    "quoteProps": "as-needed",
10   "semi": true,
11   "singleQuote": true,
12   "tabWidth": 4,
13   "trailingComma": "all",
14   "useTabs": false
15 }
```

# Scripts

- Dans package.json

```
"scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1",  
    "prepare": "husky install",  
    "eslint": "eslint './src/**\\\"',  
    "eslint:fix": "npm run eslint --fix",  
    "commit": "npx git-cz --disable-emoji",  
    "build": "tsc -p .",  
    "license-report": "license-report --output=csv --csvHeaders > ./source/_static/files/LICENSE-REPORT.csv",  
    "license": "license-checker --csv --production --out ./source/_static/files/LICENSE.csv",  
    "release": "standard-version",  
    "patch": "npm run release --release-as patch",  
    "minor": "npm run release --release-as minor",  
    "major": "npm run release --release-as major",  
    "amend": "git commit --amend --no-edit",  
    "push": "git push --follow-tags origin main",  
    "node": "node ./build/server/webhook.js",      You, 1 second ago • Uncommitted changes  
    "nodehttp": "node ./build/client/httprequest.js"  
},
```

- Execution avec : npm run nomduscript

# Programme

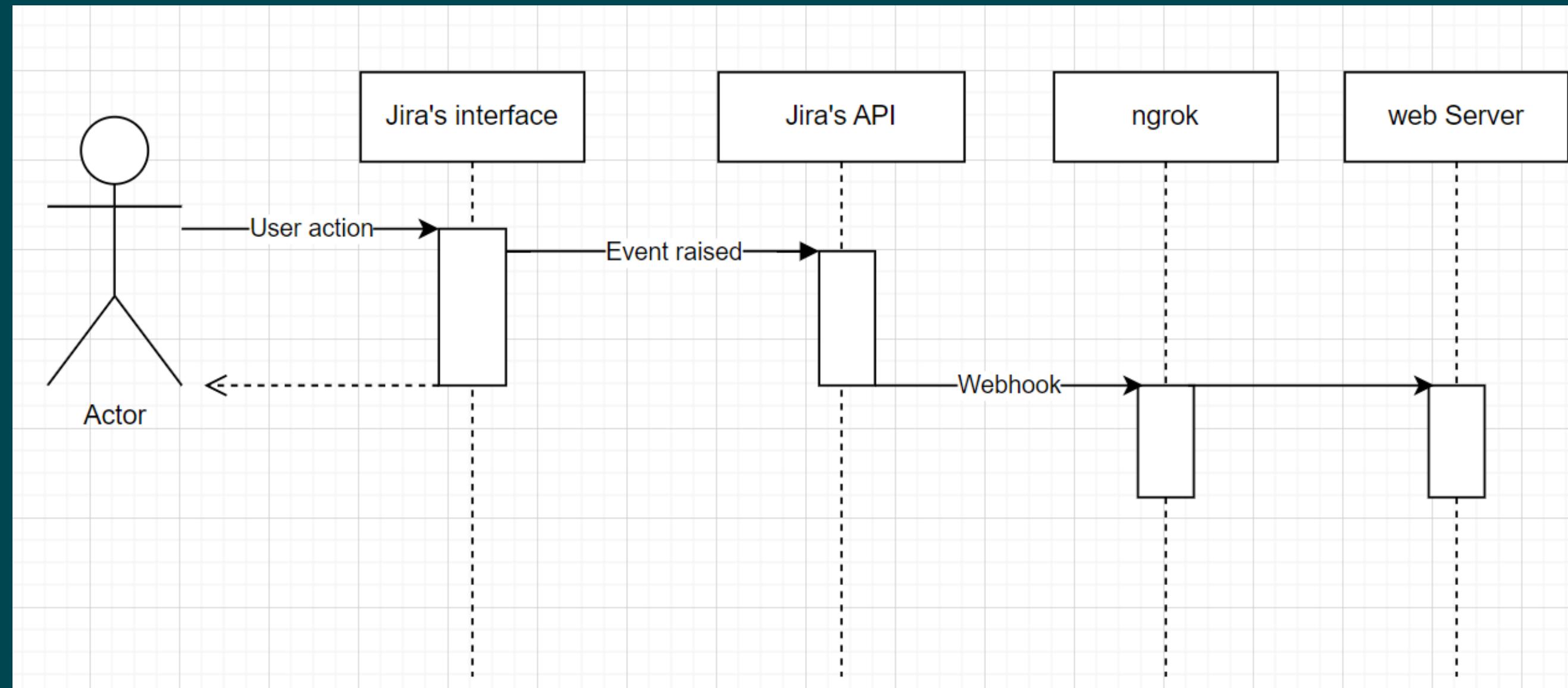
- Extensions
- Packages
- WebHook
- HTTP requests

Avec Jira et Visual Studio Code

# Webhook

Web Server : localhost by Express

API : interface de programmation d'application

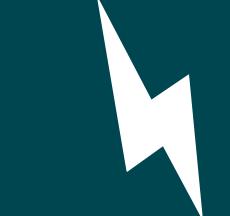


# Webhook

lien de rappel HTTP

CRUD

Jira's API



Webhook

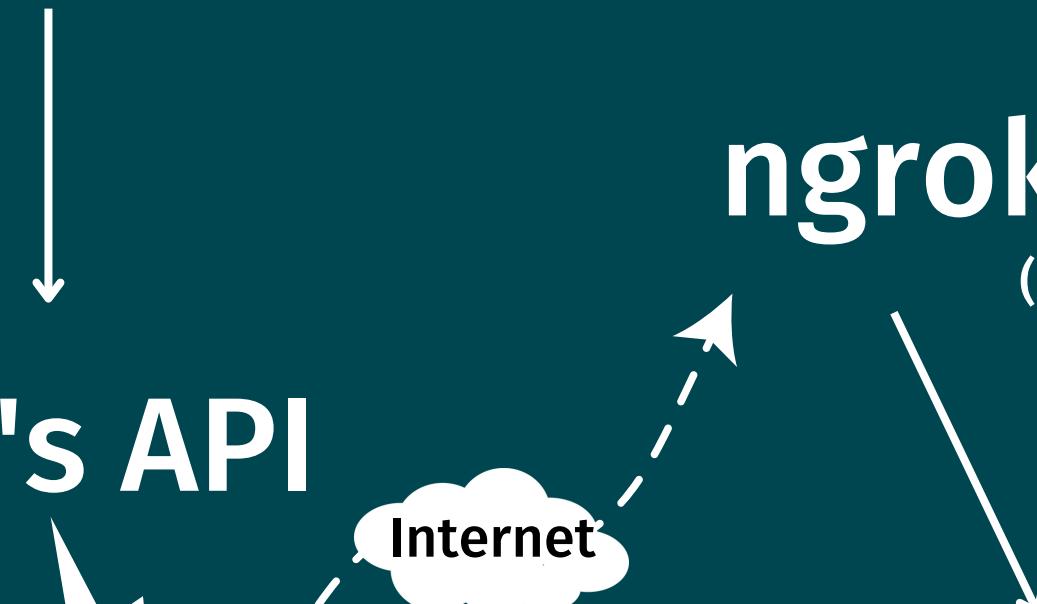
scope : • user  
• issues  
• comments

ngrok

(Ip publique)

Web Server

(localhost)



# ngrok



## héberge un serveur en ligne

```
PS C:\WINDOWS\system32> ngrok http 3000
```

```
ngrok
(Ctrl+C to quit)

Announcing ngrok-rs: The ngrok agent as a Rust crate: https://ngrok.com/rust

Session Status          online
Account                 mourad (Plan: Free)
Version                3.2.2
Region                 Europe (eu)
Latency                26ms
Web Interface          http://127.0.0.1:4040
Forwarding             https://1704-213-214-47-218.ngrok-free.app -> http://localhost:3000

Connections            ttl     opn      rt1      rt5      p50      p90
                        0       0       0.00    0.00    0.00    0.00
```

Source : <https://ngrok.com/download>

RY-SYSTEMS WFFK 3 - Presentat... ECAM2023 - STAGE - Queues +

mourad.atlassian.net/jira/servicedesk/projects/I...

OFF

Microsoft Office Ac... Boîte de réception... Claco.ecam.be Horaires ECAM Films - To Do

Your work Projects Filters More +

Need more ways to deliver service? Get a free upgrade to Jira Service Management Standard for...

Projects / ECAM2023 - STAGE / Queues

All open

1 request

| T | Key    | Summary           | Reporter        | Assignee        |
|---|--------|-------------------|-----------------|-----------------|
| ? | ECAM-5 | présentation week | Mourad Mettouqi | Mourad Mettouqi |

Administrator : Windows PowerShell

(Ctrl+C to quit)

ngrok

Session Status: online

Account: mourad (Plan: Free)

Version: 3.2.2

Region: Europe (eu)

Latency: 29ms

Web Interface: http://127.0.0.1:4848

Forwarding: https://6671-81-164-82-169.ngrok-free.app -> http://localhost:3000

Connections:

| ttl | opn | rt1  | rt5  | p50  | p90  |
|-----|-----|------|------|------|------|
| 2   | 0   | 0.00 | 0.00 | 5.08 | 5.10 |

HTTP Requests

POST / 200 OK

POST / 200 OK

Untitled (Workspace)

TS webhook.ts X

Progra Stage > stage2023project > src > server > TS webhookts > app.post() callback

64 ren.hbdv.workspace.author.displayName +

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

```
node + ...
```

votes: [Object]

}

},

changelog: { id: '10170', items: [ [Object] ] }

}

{"level": "info", "message": "issue updated: 10020 summary: 'présentation week' description: 'Jira webhook typescript'"}

Terminer le programme de commandes (O/N) ? o

PS C:\Progra Stage\stage2023project> npm run Node

> stage2023project@1.0.1 Node

> tsc -p . & node ./build/server/webhook.js

Serveur démarré sur le port 3000

main\* 0 0 ▲ 0 In 22, Col 6 Spaces: 2 UTF-8 CR/LF {} TypeScript Go Live

# Programme

- Extensions
- Packages
- WebHook
- HTTP requests

Avec Jira

# HTTP Requests

- OAuth 2.0 : Requêtes
  - Access Token
  - cloud id
- Jira : Requêtes HTTP
  - users
  - organizations
  - customers

# HTTP Requests

- Requêtes avec OAuth 2.0
  - Récupérer l'access token
  - Récupérer le cloud id

```
export interface OAuth2 {  
  clientId: string|undefined;  
  clientSecret: string|undefined;  
  cloudId?: string;  
  accessToken?: string;  
  expiresIn?: number;  
}
```

```
const auth: OAuth2 = {  
  clientId: process.env.JIRA_CLIENT_ID,  
  clientSecret: process.env.JIRA_CLIENT_SECRET  
};
```

```
import { OAuth2 } from "source/library/jira/dataModels/authentication";
```

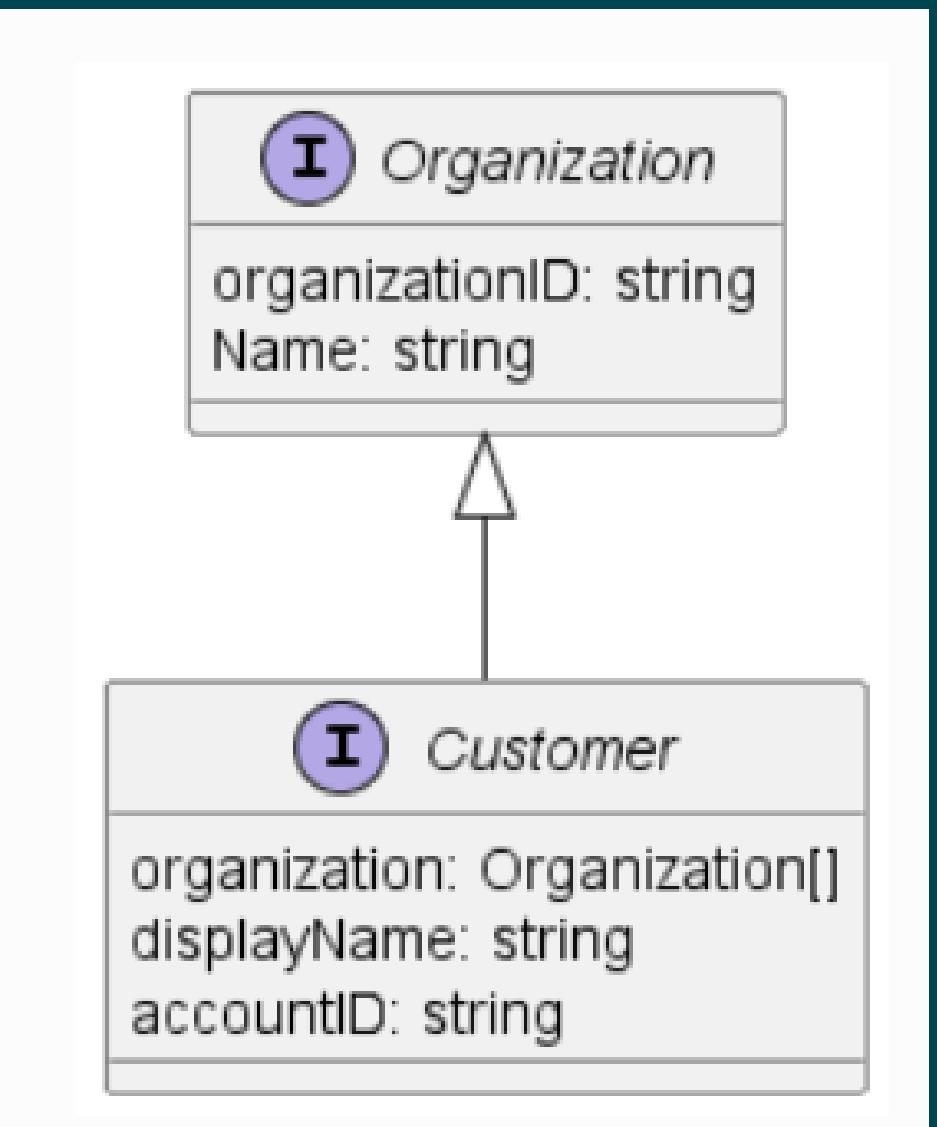
# HTTP Requests

- Requête pour récupérer l'access token
- Utilisation de `async/await`

```
// Obtenir un nouveau token d'accès à partir des "Client Credentials"
(async (): Promise<void> => {
  const responseAccessToken: AxiosResponse = await axios.post("https://api.atlassian.com/oauth/token", null, {
    params: {
      grant_type: "client_credentials",
      client_id: auth.clientId,
      client_secret: auth.clientSecret,
    },
  });
  auth.expiresIn = responseAccessToken.data.expires_in;
  auth.accessToken = responseAccessToken.data.access_token;
```

# HTTP Requests

- Interface Customer et Organization



# HTTP Requests

- Organisations : requête

```
export interface organization {  
    id: number;  
    name: string;  
    link: string;  
}
```

```
//! Request : Get Organizations  
const responseOrganisation: AxiosResponse = await axios.get(  
    `https://api.atlassian.com/ex/jira/${auth.cloudId}/rest/servicedeskapi/organization`,  
    {  
        headers: {  
            Authorization: `Bearer ${auth.accessToken}`,  
        },  
    }  
);  
// Boucle pour définir les propriétés de chaque organisation puis les envoyer dans mon array d'organisations  
for (let i = 0; i < responseOrganisation.data.values.length; i++) {  
    let orga: organization = {  
        id: responseOrganisation.data.values[i].id,  
        name: responseOrganisation.data.values[i].name,  
        link: responseOrganisation.data.values[i]._links.self  
    };  
    organizations.push(orga);  
}  
console.log("Organisations : ");  
console.log(organizations);
```

# HTTP Requests

- Organisations : réponse (JSON)

```
Organisations :  
[  
  {  
    id: '1',  
    name: 'ECAM',  
    link: 'https://api.atlassian.com/ex/jira/8a5f5ebc-5cf2-46b1-bb77-f8f57ec5e7ad/rest/servicedeskapi/organization/1'  
  },  
  {  
    id: '2',  
    name: 'BYSYSTEMS',  
    link: 'https://api.atlassian.com/ex/jira/8a5f5ebc-5cf2-46b1-bb77-f8f57ec5e7ad/rest/servicedeskapi/organization/2'  
  },  
  {  
    id: '3',  
    name: 'conrentin',  
    link: 'https://api.atlassian.com/ex/jira/8a5f5ebc-5cf2-46b1-bb77-f8f57ec5e7ad/rest/servicedeskapi/organization/3'  
  }]
```

# HTTP Requests

- Customers : interface

```
import { organization } from "./organization";  
  
You, 23 hours ago | 1 author (You)  
• export interface customer {  
    accountId: string;  
    name: string;  
    organizationId: organization[];  
}
```

# HTTP Requests

- Customers : requête

```
//! Request : Get users in organization
const responseOrganisationUser: AxiosResponse = await axios.get(
  `https://api.atlassian.com/ex/jira/${auth.cloudId}/rest/servicedeskapi/organization/${organizations[i].id}/user`,
  {
    headers: {
      Authorization: `Bearer ${auth.accessToken}`,
    },
  }
);
// Boucle pour définir les propriétés de chaque user de l'organisation
for (let j = 0; j < responseOrganisationUser.data.values.length; j++) {
  // si le customer existe déjà, ne va pas le réinstancier
  const found = customers.find((obj) => {
    return obj.name === responseOrganisationUser.data.values[j].displayName;
  });
  if (found) {
    found.organizationId.push(organizations[i]);
    console.log(found);
  } else {
    let cust: customer = {
      accountId: responseOrganisationUser.data.values[j].accountId,
      name: responseOrganisationUser.data.values[j].displayName,
      organizationId: [],
    };
    cust.organizationId.push(organizations[i]);
    customers.push(cust);
    console.log(cust);
  }
}
```

# HTTP Requests

- Customers : réponse (JSON)

```
Customers :  
[  
  {  
    accountId: 'qm:0161d951-9950-4dc6-91ab-6beebd467e12:a3071500-0e1d-4d2c-8513-651a66304413',  
    name: 'MaillardCorentin',  
    organizationId: [ [Object], [Object] ]  
  },  
  {  
    accountId: 'qm:0161d951-9950-4dc6-91ab-6beebd467e12:a65b9a8d-81fc-4b21-92ad-02f4a156bdcb',  
    name: 'Boujraf Youssef',  
    organizationId: [ [Object] ]  
  },  
  {  
    accountId: 'qm:0161d951-9950-4dc6-91ab-6beebd467e12:de6e2ebc-f53f-40e3-ae16-0e4339ca4e86',  
    name: 'Mourad',  
    organizationId: [ [Object] ]  
  }  
]
```

# Avez-vous des questions ?

N'hésitez pas à nous contacter !



Van Olffen Victor  
20253@ecam.be



Maillard Corentin  
21306@ecam.be



Mettioui Mourad  
195019@ecam.be

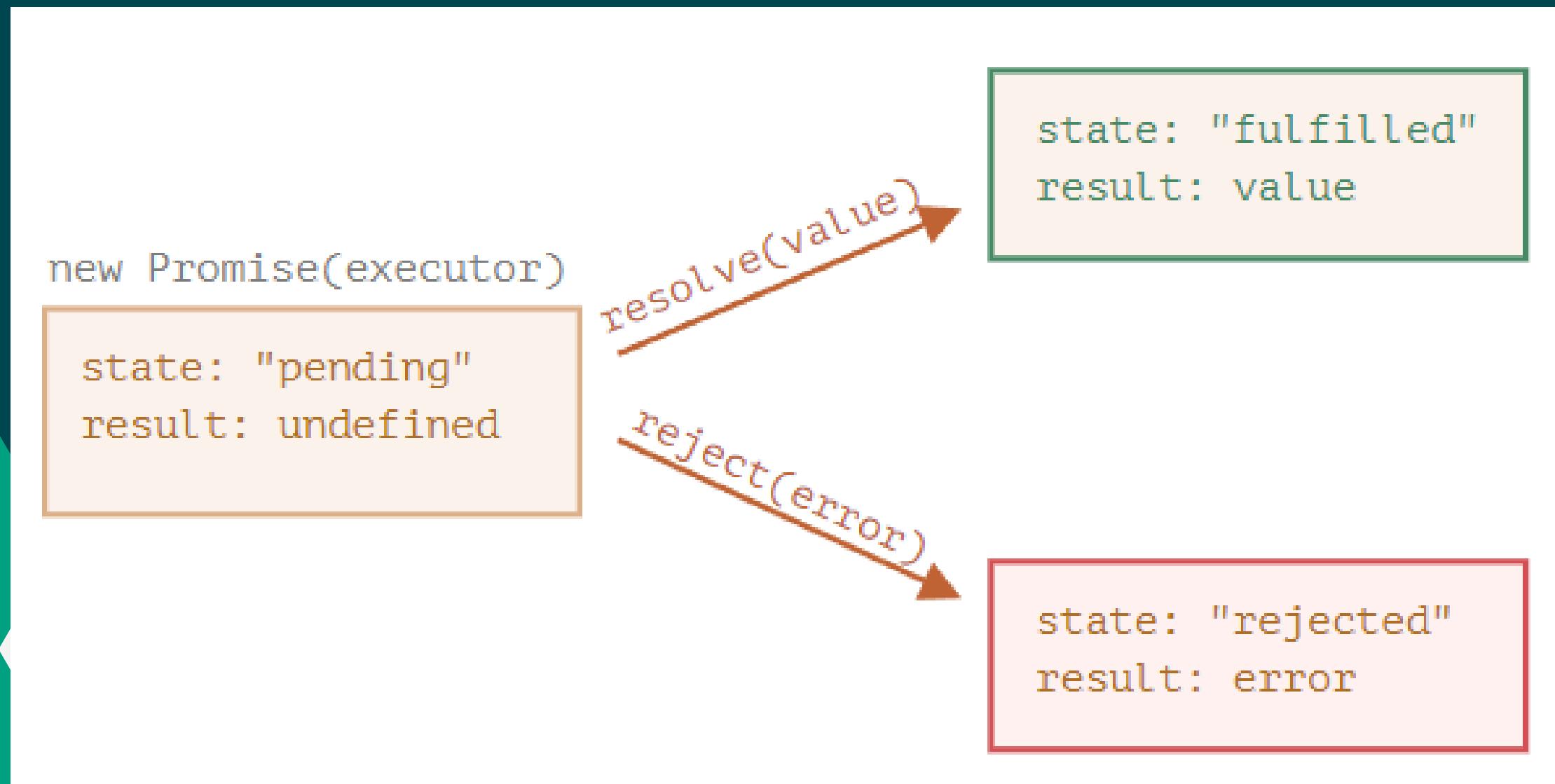
# Promise

## Utilité

- Permet de continuer d'exécuter le code
- Dire quand elle donnera ça réponse
- Donne la promesse au partie du code qui en ont besoin

# Promise

- 3 états :



# Promise

.then

- Promise résolue donne sa réponse

.finally

- Plus nécessaire => cleanup

# Postman



- Requête avec OAuth 2.0

This screenshot shows the 'Authorization' tab in Postman's configuration interface. The tab is highlighted with a green dot. Below the tabs are several descriptive text blocks and dropdown menus.

- Type:** OAuth 2.0
- Description:** This authorization method will be used for every request in this collection. You can override it on a per-request basis.
- Add auth data to:** Request Headers
- Note:** The authorization data will be automatically generated when you send the request. Learn more about [authorization ↗](#)

# Postman



- Crédit d'un nouveau token
- Access Token URL :  
<https://auth.atlassian.com/oauth/token>

Configure New Token

Configuration Options • Advanced Options

Token Name: JIRA

Grant Type: Client Credentials

Access Token URL: <https://auth.atlassian.com/oauth/tok...>

Client ID: {{clientID}}

Client Secret: {{clientSecret}}

Scope: e.g. read:org

Client Authentication: Send as Basic Auth header

Clear cookies

**Get New Access Token**

# Postman



- Requête pour récupérer le Cloud ID

GET <https://api.atlassian.com/oauth/token/accessible-resources> Send

```
[  
  {  
    "id": "332a61e4-de76-4c4b-a218-3f00bc6b47a3",  
    "url": "https://corentinmaillard.atlassian.net",  
    "name": "corentinmaillard",  
    "scopes": [  
      "read:servicedesk-request",  
      "manage:servicedesk-customer",  
      "write:servicedesk-request"  
    ],  
    "avatarUrl": "https://site-admin-avatar-cdn.prod.public.atl-paas.net/avatars/240/pencilmarker.png"  
  }]  
]
```

# API token

- Procédure

# API token

- Créer un API token sur Atlassian



**Créer un jeton d'API**

Choisissez une étiquette courte et facile à retenir.

Label \*

En créant un jeton d'API, vous acceptez les [Conditions d'utilisation des développeurs Atlassian](#) et acceptez la [Politique de confidentialité](#).

Annuler      **Créer**

Source : <https://id.atlassian.com/manage-profile/security/api-tokens>

# Postman



- Utilisation API token

The screenshot shows the Postman interface with the 'Authorization' tab selected. The 'Type' dropdown is set to 'Basic Auth'. A note in the sidebar cautions about the security of using basic auth in a collaboration environment and recommends using variables. The 'Username' field contains the variable {{atlassian\_email}} and the 'Password' field contains the variable {{atlassian\_token}}. A 'Show Password' checkbox is checked.

Params • Authorization • Headers (9) Body Pre-request Script Tests Settings

Type Basic Auth

(i) Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaboration environment, we recommend using variables. Learn more about [variables](#) ↗

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#) ↗

Username {{atlassian\_email}}

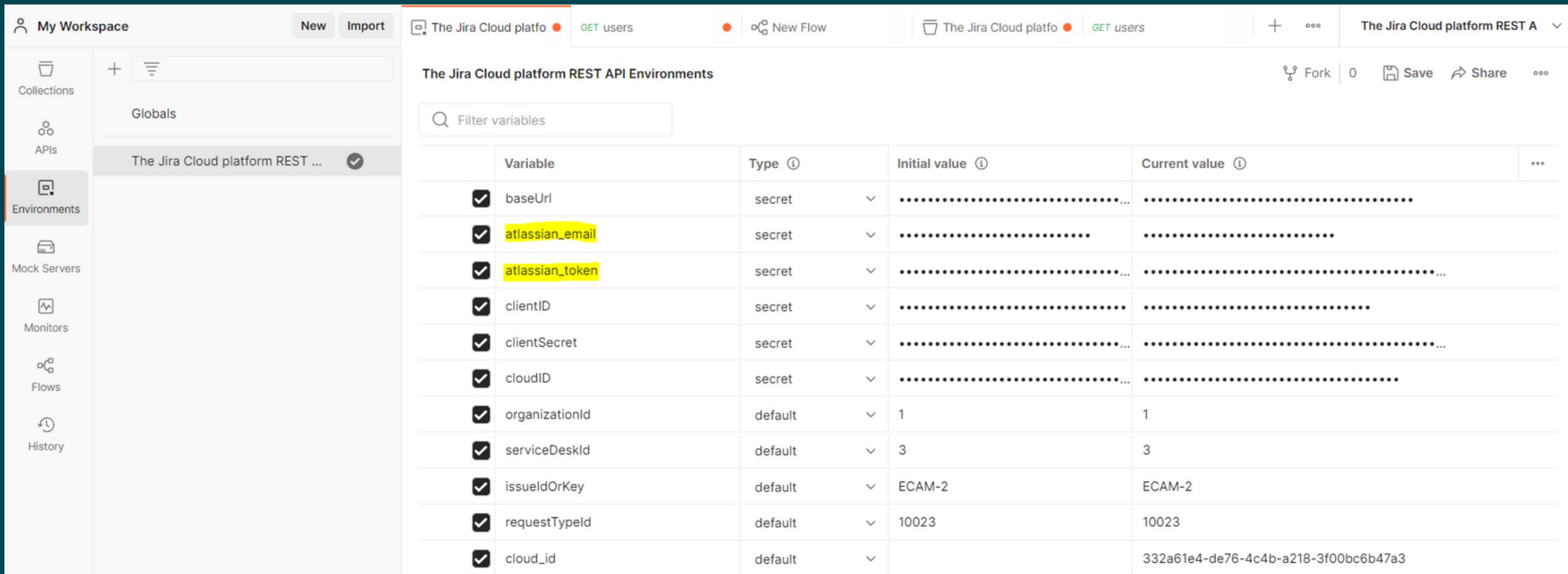
Password {{atlassian\_token}}

Show Password

# Postman



- Utilisation d'environnements



The screenshot shows the Postman interface with the "Environments" tab selected in the sidebar. The main area displays a table of global variables for the "The Jira Cloud platform REST API Environments".

|                                     | Variable        | Type    | Initial value | Current value                        |
|-------------------------------------|-----------------|---------|---------------|--------------------------------------|
| <input checked="" type="checkbox"/> | baseUrl         | secret  | .....         | .....                                |
| <input checked="" type="checkbox"/> | atlassian_email | secret  | .....         | .....                                |
| <input checked="" type="checkbox"/> | atlassian_token | secret  | .....         | .....                                |
| <input checked="" type="checkbox"/> | clientID        | secret  | .....         | .....                                |
| <input checked="" type="checkbox"/> | clientSecret    | secret  | .....         | .....                                |
| <input checked="" type="checkbox"/> | cloudID         | secret  | .....         | .....                                |
| <input checked="" type="checkbox"/> | organizationId  | default | 1             | 1                                    |
| <input checked="" type="checkbox"/> | serviceDeskId   | default | 3             | 3                                    |
| <input checked="" type="checkbox"/> | issueIdOrKey    | default | ECAM-2        | ECAM-2                               |
| <input checked="" type="checkbox"/> | requestTypeId   | default | 10023         | 10023                                |
| <input checked="" type="checkbox"/> | cloud_id        | default |               | 332a61e4-de76-4c4b-a218-3f00bc6b47a3 |

# OAuth 1.0 et 2.0

- Version majeur (réécriture complète)  
=> 2.0 n'est pas rétrocompatible avec 1.0

|                              | OAuth 1.0       | OAuth 2.0                        |
|------------------------------|-----------------|----------------------------------|
| Supporte                     | web workflows   | web workflows<br>non-web clients |
| Durée<br>access token        | 1 years or more | 1 hour                           |
| Clear séparation<br>of roles | No              | Yes                              |

# OAuth 2.0

- Procédure

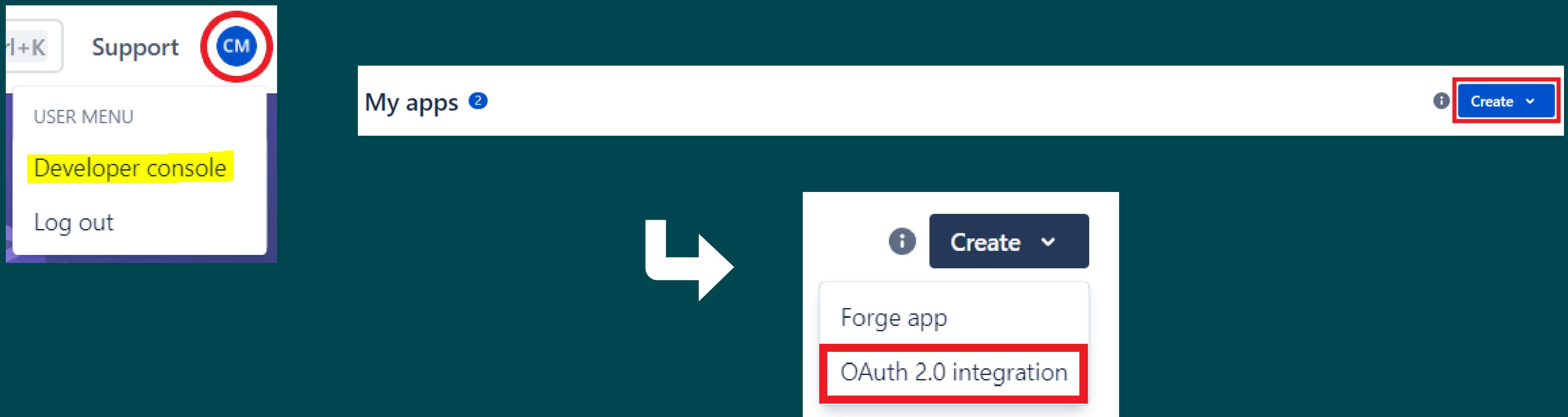
# Atlassian Developer



- Personnalisez les produits Atlassian
- Créer des applications

# Atlassian Developer

- Créer une app avec OAuth 2.0



Source : <https://developer.atlassian.com/>

# Atlassian Developer

- Creation OAuth 2.0

## Create a new OAuth 2.0 (3LO) integration

An app provides API credentials for Atlassian products and services, as well as features such as OAuth 2.0 (3LO).

Name \*

Postman

Name your app according to its purpose, for example, Dropbox integration or Timesheets for Jira.



I agree to be bound by [Atlassian's developer terms](#).

Create

Cancel

# Atlassian Developer



- Postman

- Ajout du Callback URL

**ATLASSIAN Developer**

POSTMAN

Overview

Distribution

Permissions

**Authorization**

Console / My apps / [Postman](#) / Authorization / **Authorization**

**OAuth 2.0 authorization code grants (3LO) for apps**

Configure OAuth 2.0 authorization code grants to allow your app to access data (within specific scopes) from Atlassian products on the user's behalf. Learn more about OAuth 2.0 authorization code grants for [Jira Cloud](#) and [Confluence Cloud](#).

**Callback URL \***

`https://oauth.pstmn.io/v1/browser-callback`

**Save changes**

jira

Authorization Code

`https://oauth.pstmn.io/v1/browser-callback`

`https://example.com/login/oauth/authorize`

`https://auth.atlassian.com/oauth/token`

`{{clientId}}`

`{{clientSecret}}`

e.g. read:org

State

# Atlassian Developer



- Configuration des permissions

A screenshot of the Atlassian developer console interface. On the left, there's a sidebar with the following options: POSTMAN, Overview, Distribution, Permissions (which is highlighted with a blue bar), Authorization, and Settings. The main area shows the 'Permissions' configuration for the 'Postman' app.

Console / My apps / Postman /

## Permissions

Add and configure your app's API scopes. See [OAuth 2.0 \(3LO\) for apps](#).

**Scopes Used**  
18

**i Scopes**  
We recommend that you don't add more than 50 scopes to your app. Use classic scopes to minimize the number of scopes you need. [Learn more](#)

| API name   | Scopes used | Action  |
|--|-------------|---|
| User identity API<br>Get the profile details for the currently logged-in user, such as the Atlassian account ID and email. | 2           | <a href="#">Config...</a> <a href="#">Documentation</a> |
| Confluence API<br>Get, create, update, and delete content, spaces, and more.   | 0           | <a href="#">Add</a> <a href="#">Documentation</a>       |
| BRIE API<br>Create, cancel, and read backup and restore, retrieve and publish cloud details.                               | 0           | <a href="#">Add</a> <a href="#">Documentation</a>       |
| Jira API<br>Get, create, update, and delete issues, projects, fields, and more.  | 16          | <a href="#">Config...</a> <a href="#">Documentation</a> |
| Personal data reporting API<br>Report user accounts that an app is storing personal data for.                              | 0           | <a href="#">Add</a> <a href="#">Documentation</a>       |

# Atlassian Developer



## • Permissions pour Jira Service Management API

Jira Service Management API Edit Scopes

| Select                              | Scope Name  | Code                                   |
|-------------------------------------|---|--|
| <input checked="" type="checkbox"/> | View Jira Service Management request data<br>Read customer request data, including approvals, attachments, comments, request participants, and status/transitions. Read service desk and request types, including searching for request types and reading request type fields, properties and groups. | read:servicedesk-request               |
| <input checked="" type="checkbox"/> | Manage Jira Service Management customers and organizations<br>Create, manage and delete customers and organizations. Add and remove customers and organizations from service desks.   | manage:servicedesk-customer            |
| <input checked="" type="checkbox"/> | Create and manage Jira Service Management requests<br>Create and edit customer requests, including add comments and attachments, approve, share (add request participants), subscribe, and transition.  | write:servicedesk-request              |
| <input checked="" type="checkbox"/> | Read Insight objects<br>Read Insight object schemas, objects, and object attributes.  | read:servicemanagement-insight-objects |

# Atlassian Developer

- Permissions pour Jira

| Select                              | Scope Name   | Code                      |
|-------------------------------------|--|---------------------------|
| <input checked="" type="checkbox"/> | View Jira issue data<br>Read Jira project and issue data, search for issues, and objects associated with issues like attachments and worklogs.     | read:jira-work            |
| <input checked="" type="checkbox"/> | Manage project settings<br>Create and edit project settings and create new project-level objects (e.g. versions and components).                   | manage:jira-project       |
| <input checked="" type="checkbox"/> | Manage Jira global settings<br>Take Jira administration actions (e.g. create projects and custom fields, view workflows, manage issue link types). | manage:jira-configuration |
| <input checked="" type="checkbox"/> | View user profiles<br>View user information in Jira that the user has access to, including usernames, email addresses, and avatars.                | read:jira-user            |
| <input checked="" type="checkbox"/> | Create and manage issues<br>Create and edit issues in Jira, post comments as the user, create worklogs, and delete issues.                         | write:jira-work           |
| <input checked="" type="checkbox"/> | Manage Jira webhooks<br>Register and manage Jira webhooks.   | manage:jira-webhook       |
| <input checked="" type="checkbox"/> | Manage development and release information for third parties in Jira<br>Manage development and release information for third parties in Jira.      | manage:jira-data-provider |

Edit Scopes

# Atlassian Developer



- Permissions  
granular scopes

Console / My apps / Postman / Permissions / **Jira API**

Add and configure your app's API scopes. See OAuth 2.0 (3LO) for apps.

Scopes Used  
**20**

**i Choosing your scopes**  
Use the scopes recommended in the API documentation for the

[Classic scopes](#) [Granular scopes](#)

# Atlassian Developer



- Permissions granular scopes

|  |   |
|--|---|
| View application roles                                     | read:application-role:jira                            |
| View audit logs  | read:audit-log:jira                                   |
| View issue worklogs  | read:issue-worklog:jira                               |
| View issue worklog properties                              | read:issue-worklog.property:jira                      |
| View users   | read:user:jira  |
| Allows the app to read customer accounts information       | read:customer:jira-service-management                 |
| Allows the app to read organisation information            | read:organization:jira-service-management             |
| Allows the app to read organisation membership information | read:organization.user:jira-service-management        |
| Allows the app to list & search service desks              | read:servicedesk:jira-service-management              |
| Allows the app to list organisations to service desks      | read:servicedesk.organization:jira-service-management |
| Allows the app to list & search request types              | read:requesttype:jira-service-management              |
| Allows the app to read request type desk entity properties | read:requesttype.property:jira-service-management     |
| Allows the app to list queues                              | read:queue:jira-service-management                    |
| Allows the app to list & search requests                   | read:request:jira-service-management                  |

# Atlassian Developer



## • Activer les permissions

The screenshot shows the Atlassian Developer API documentation page. The 'Authorization' section is highlighted with a blue background. Other sections like 'POSTMAN', 'Overview', 'Distribution', 'Permissions', and 'Settings' are visible but not highlighted.

### Authorization URL generator i

Use this URL to install your 3LO app.

#### User identity API authorization URL

```
https://auth.atlassian.com/authorize?audience=api.atlassian.com&client_id=nr3DZPJ129vipN5AwMPBiAk0AZDBls0I&scope=read%3Ame%20read%3Aaccount&redirect_uri=https%3A%2F%2Foauth.pstmn.io%2Fv1%2Fbrowser-callba
```



#### Classic Jira Service Management API authorization URL

```
ead%3Aservicemanagement-insight-objects&redirect_uri=https%3A%2F%2Foauth.pstmn.io%2Fv1%2Fbrowser-callb&ack&state=${YOUR_USER_BOUND_VALUE}&response_type=code&prompt=consent
```



#### Classic Jira platform REST API authorization URL

```
https://auth.atlassian.com/authorize?audience=api.atlassian.com&client_id=nr3DZPJ129vipN5AwMPBiAk0AZDBls0I&scope=read%3Ajira-work%20manage%3Ajira-project%20read%3Ajira-user%20manage%3Ajira-configuration%20
```

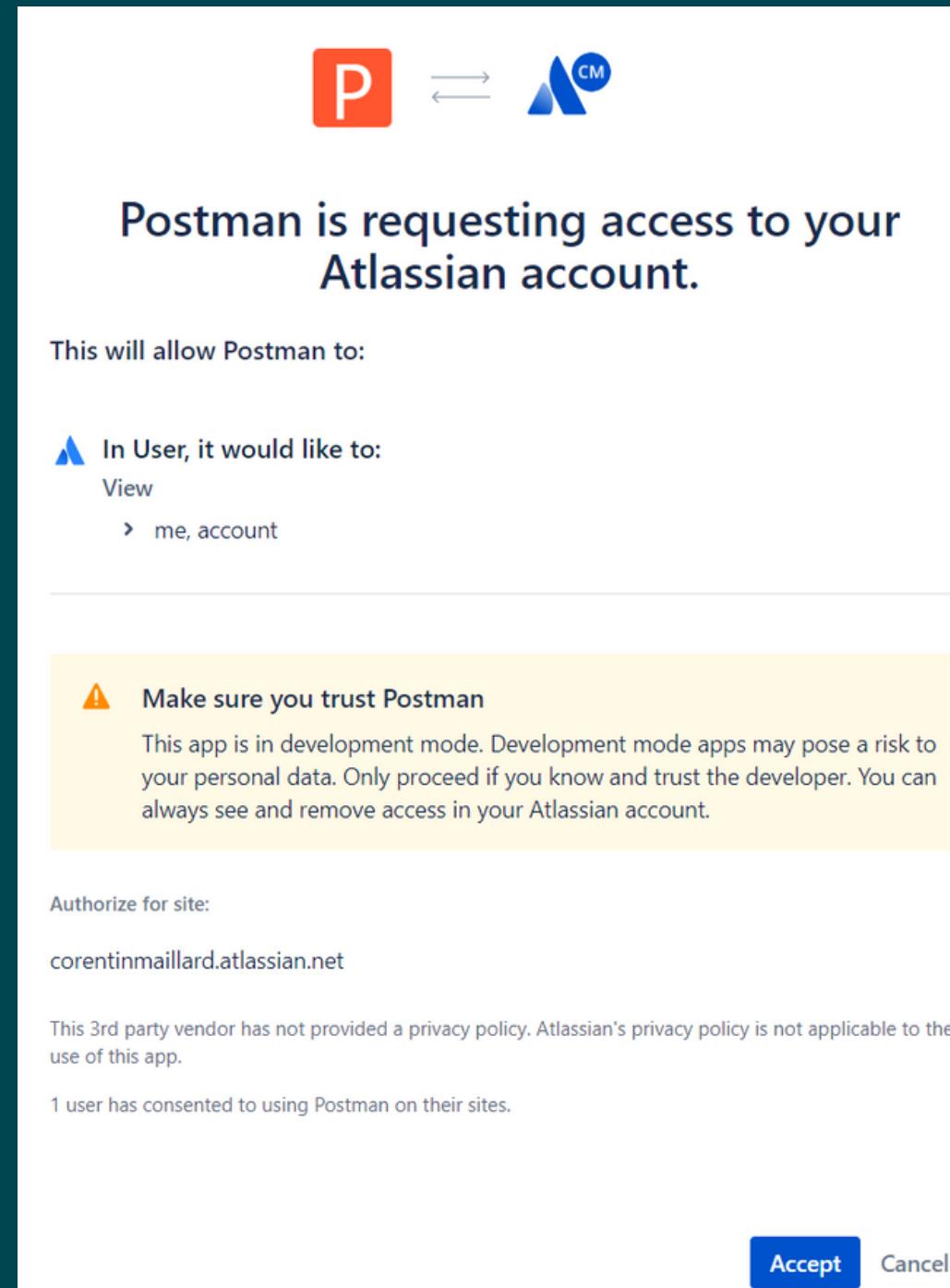


#### Granular Jira API authorization URL

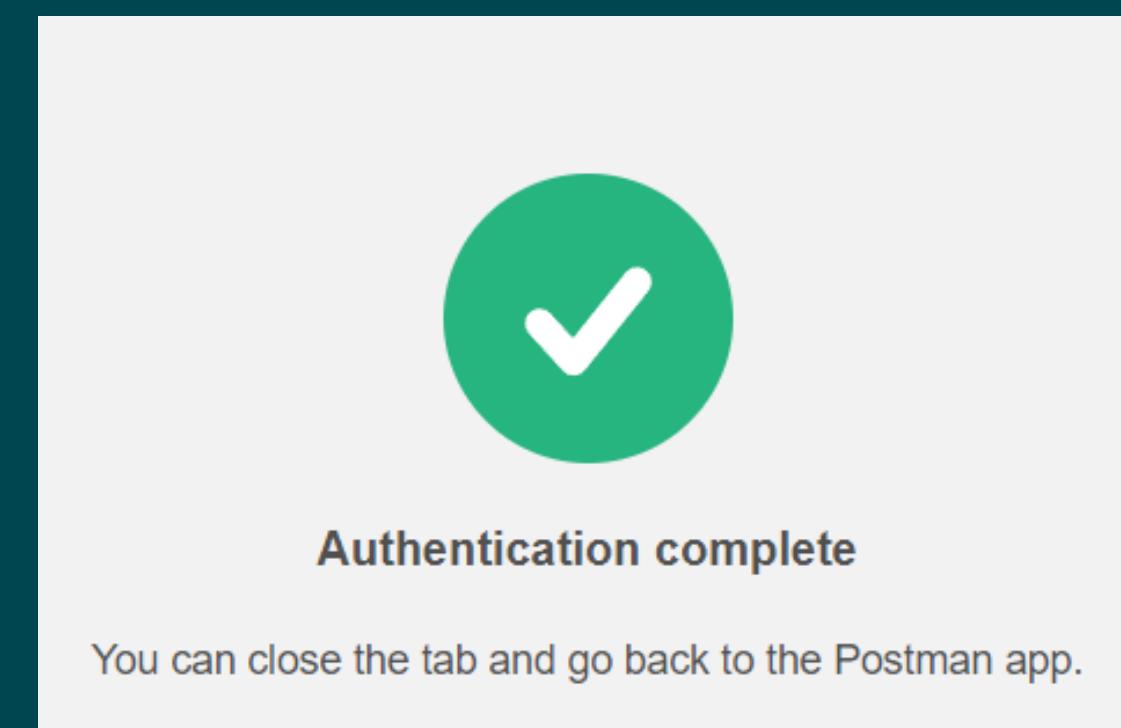
```
https://auth.atlassian.com/authorize?audience=api.atlassian.com&client_id=nr3DZPJ129vipN5AwMPBiAk0AZDBls0I&scope=read%3Aapplication-role%3Ajira%20read%3Auser%3Ajira%20read%3Aaudit-log%3Ajira%20read%3Aorgan
```



# Atlassian Developer



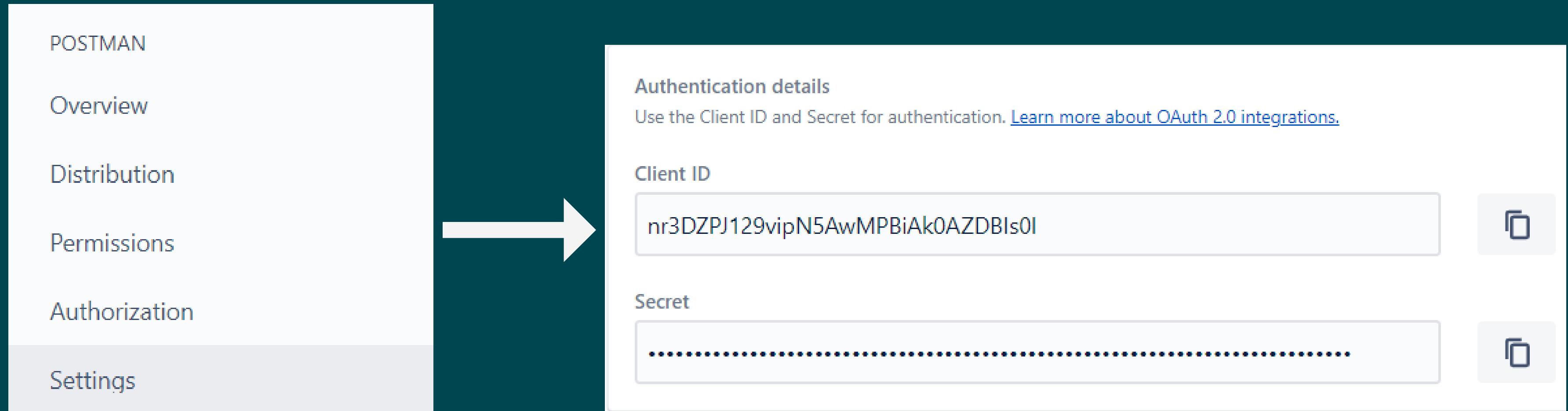
## • Activer les permissions



# Atlassian Developer



- Client ID et Client Secret



# Postman

Application pour tester des APIs



Système d'exploitation : Windows 10/11

# Postman



- Autorisation OAuth 2.0

This screenshot shows the 'Authorization' tab in Postman's configuration interface. The tab is highlighted with a green dot. Below the tabs, a descriptive message states: 'This authorization method will be used for every request in this collection. You can override this at the request level.' Under the 'Type' section, 'OAuth 2.0' is selected from a dropdown menu. A note below explains: 'The authorization data will be automatically generated when you send the request. Learn more about [authorization](#)'.

Authorization • Pre-request Script Tests Variables Runs

This authorization method will be used for every request in this collection. You can override this at the request level.

Type OAuth 2.0

The authorization data will be automatically generated when you send the request. Learn more about [authorization](#) ↗

Add auth data to Request Headers

# Postman



- Crédation d'un nouveau token

Configure New Token

Configuration Options ● Advanced Options

|                       |   |
|-----------------------|---|
| Token Name            | JIRA  |
| Grant Type            | Client Credentials <span style="float: right;">▼</span>                                     |
| Access Token URL ⓘ    | <a href="https://auth.atlassian.com/oauth/token">https://auth.atlassian.com/oauth/token</a> |
| Client ID ⓘ           | nr3DZPJ129vipN5AwMPBiAk0AZDB⚠   |
| Client Secret ⓘ       | ATOAA7cwLZ80GoDvHTBInlh-IJgK⚠   |
| Scope ⓘ               | e.g. read:org   |
| Client Authentication | Send as Basic Auth header <span style="float: right;">▼</span>                              |

● Clear cookies ⓘ

Get New Access Token

# Postman



- Cloud ID

The screenshot shows the Postman interface with a GET request to <https://api.atlassian.com/oauth/token/accessible-resources>. The response body is displayed below:

```
[  
  {  
    "id": "332a61e4-de76-4c4b-a218-3f00bc6b47a3",  
    "url": "https://corentinmaillard.atlassian.net",  
    "name": "corentinmaillard",  
    "scopes": [  
      "read:servicedesk-request",  
      "manage:servicedesk-customer",  
      "write:servicedesk-request"  
    ],  
    "avatarUrl": "https://site-admin-avatar-cdn.prod.public.atl-paas.net/avatars/240/pencilmarker.png"  
  }]  
]
```

# Postman



- Commande avec Méthode OAuth 2.0 :

`https://api.atlassian.com/ex/jira/{{cloudID}}/{{api}}`

# Postman



- Exemple :

```
GET https://api.atlassian.com/ex/jira/{{cloudID}}/rest/api/3/users/search?startAt=0&maxResults=50
```

# Postman



The screenshot shows the Postman application interface. The left sidebar contains 'My Workspace' collections, APIs, environments, mock servers, monitors, flows, and history. The main workspace shows a collection named 'JIRA oauth 2.0 / get user'. A specific request is selected: 'GET https://api.atlassian.com/ex/jira/[cloud\_45]/rest/api/3/users/search?startAt=0&maxResults=50'. The 'Authorization' tab is active, showing 'OAuth 2.0' selected as the type, with a token named 'jira cloud' containing the value 'eyJraWQiOiUmZTM2ZThkMzZjMTA2N2RjY...'. The 'Request Headers' section includes 'Header Prefix' set to 'Bearer'. Below the authorization, there's an 'Auto-refresh token' option and a 'Share token' option. The 'Body' tab shows a JSON response with two user objects. The first user has an account ID of '712820:e04b5e29-1c39-4d00-a9bc-c50899a22274', a display name of 'Mouzed Mettoui', and is active. The second user has an account ID of '5b70c0b80fd0ac05d3095a09', a display name of 'Chat Notifications', and is also active. Both users have multiple avatar URLs for different sizes (24x24, 16x16, 32x32). The status bar at the bottom indicates 'Status: 200 OK Time: 695 ms Size: 26.8 KB'.

```
[{"self": "https://api.atlassian.com/ex/jira/3e349891-1034-4d3a-900d-75b56a39f784/rest/api/3/user?accountId=712820:e04b5e29-1c39-4d00-a9bc-c50899a22274", "accountId": "712820:e04b5e29-1c39-4d00-a9bc-c50899a22274", "accountType": "atlassian", "avatarsUrls": {"48x48": "https://secure.gravatar.com/avatar/131bc91ba141d45a2095540f051932c7d+https%3A%2F%2Favatars-management.appspot.us-west-2.prod.public.sti-paas.net%2Finitials%2FMM-0.png", "24x24": "https://secure.gravatar.com/avatar/131bc91ba141d45a2095540f051932c7d+https%3A%2F%2Favatars-management.appspot.us-west-2.prod.public.sti-paas.net%2Finitials%2FMM-0.png", "16x16": "https://secure.gravatar.com/avatar/131bc91ba141d45a2095540f051932c7d+https%3A%2F%2Favatars-management.appspot.us-west-2.prod.public.sti-paas.net%2Finitials%2FMM-0.png", "32x32": "https://secure.gravatar.com/avatar/131bc91ba141d45a2095540f051932c7d+https%3A%2F%2Favatars-management.appspot.us-west-2.prod.public.sti-paas.net%2Finitials%2FMM-0.png"}, "l", "displayName": "Mouzed Mettoui", "active": true, "locale": "en_US"}, {"self": "https://api.atlassian.com/ex/jira/3e349891-1034-4d3a-900d-75b56a39f784/rest/api/3/user?accountId=5b70c0b80fd0ac05d3095a09", "accountId": "5b70c0b80fd0ac05d3095a09", "accountType": "app", "avatarsUrls": {"48x48": "https://secure.gravatar.com/avatar/7d674fbf09569d0bd3f2c19ae5b4462a7d+https%3A%2F%2Favatars-management.appspot.us-west-2.prod.public.sti-paas.net%2Finitials%2FCN-5.png", "24x24": "https://secure.gravatar.com/avatar/7d674fbf09569d0bd3f2c19ae5b4462a7d+https%3A%2F%2Favatars-management.appspot.us-west-2.prod.public.sti-paas.net%2Finitials%2FCN-5.png", "16x16": "https://secure.gravatar.com/avatar/7d674fbf09569d0bd3f2c19ae5b4462a7d+https%3A%2F%2Favatars-management.appspot.us-west-2.prod.public.sti-paas.net%2Finitials%2FCN-5.png", "32x32": "https://secure.gravatar.com/avatar/7d674fbf09569d0bd3f2c19ae5b4462a7d+https%3A%2F%2Favatars-management.appspot.us-west-2.prod.public.sti-paas.net%2Finitials%2FCN-5.png"}, "l", "displayName": "Chat Notifications", "active": true}, {"self": "https://api.atlassian.com/ex/jira/3e349891-1034-4d3a-900d-75b56a39f784/rest/api/3/user?accountId=557058:cfc04d7b-be84-4be4-e567aa5332c6"}]
```

# Webhook

- Procédure

# Webhook

lien de rappel HTTP



# Webhook



- Postman

The screenshot shows the Postman Flow builder interface. On the left, there's a sidebar with categories: Collections, APIs, Environments, Mock Servers, Monitors, Flows (which is highlighted with a red box and labeled 1), and History. In the main area, there's a 'My Workspace' section with a '+ New Flow' button (labeled 2) and a 'webhook' flow card. Above the workspace, there are buttons for 'New', 'Import', 'webhook', 'New Flow' (which is also highlighted with a red box and labeled 3), and a search bar. A large text box in the center says 'New Flow'. At the bottom, there's a toolbar with 'Start' (with a gear icon), 'Aa' (with a font icon), and a 'Click and drag to add new blocks' instruction with a curved arrow pointing to the toolbar.

# Webhook



- Postman

The screenshot shows the Postman application interface. The top navigation bar includes Home, Workspaces, API Network, Explore, a search bar, and various user and system icons. The left sidebar, titled "My Workspace", contains sections for Collections, APIs, Environments, Mock Servers, Monitors, Flows (which is selected), and History. Under the Flows section, there is a "New Flow" card labeled "webhook". The main workspace is titled "New Flow" and contains a large, empty grid area for defining the flow logic. At the bottom of the workspace, there are buttons for "Start", "Run", and "Block". The bottom navigation bar includes links for Online, Find and Replace, Console, Cookies, Capture requests, Runner, and Trash.

# Webhook



- Accès à webhook à partir de Jira

The screenshot shows the Atlassian Home page. At the top left is the Atlassian logo with a red circle around it. The top navigation bar includes 'Home' and 'Notifications'. A banner for 'team '23' is displayed, followed by a message: 'See how Jira, Confluence, and Trello can help your team achieve its biggest goals together at Team'. Below this is a 'Switch to' section with icons for various Atlassian products:

| Icon                         | Product                 | User   |
|------------------------------|-------------------------|--------|
| Confluence icon              | Confluence              | mourad |
| Jira Work Management icon    | Jira Work Management    | mourad |
| Jira Service Management icon | Jira Service Management | mourad |
| Jira Software icon           | Jira Software           | mourad |
| Atlas icon                   | Atlas                   | TRY    |
| Administration icon          | Administration          |        |
| Account settings icon        | Account settings        |        |

# Webhook



- Accès à webhook à partir de Jira

The screenshot shows the Atlassian Home page. At the top left is the Atlassian logo with a red circle around it. Below the logo are navigation links for "Home" and "Notifications". A banner for "team '23" is displayed, followed by a message: "See how Jira, Confluence, and Trello can help your team achieve its biggest goals together at Team". Below the banner is a "Switch to" section with icons and names for various Atlassian products:

| Product                 | User   |
|-------------------------|--------|
| Confluence              | mourad |
| Jira Work Management    | mourad |
| Jira Service Management | mourad |
| Jira Software           | mourad |
| Atlas                   | TRY    |
| Administration          |        |
| Account settings        |        |



# Webhook



- Accès à webhook à partir de Jira

The screenshot shows the Atlassian Team '23 dashboard. At the top left is the Atlassian logo with a red circle around it. Below the logo is a banner for 'team' 23. The main area has a purple header with the text 'See how Jira, Confluence, and Trello can help your team achieve its biggest goals together at Team'. Below this are several product links:

- Confluence (mourad)
- Jira Work Management (mourad)
- Jira Service Management (mourad)
- Jira Software (mourad)
- Atlas (TRY)
- Administration
- Account settings



The screenshot shows the Jira Administration interface. At the top, there is a search bar with 'Find by name' and filters for 'Products' and 'Plans'. Below this, a message says 'Showing 4 results out of 4 products'.

| Product  | Plan | Users |
|--|------|-------|
| Confluence<br><a href="https://mourad.atlassian.net/wiki">https://mourad.atlassian.net/wiki</a>    | Free | 1     |
| Jira Administration<br><a href="https://mourad.atlassian.net">https://mourad.atlassian.net</a>     |      |       |
| Jira Service Management<br><a href="https://mourad.atlassian.net">https://mourad.atlassian.net</a> |      |       |
| Jira Software<br><a href="https://mourad.atlassian.net">https://mourad.atlassian.net</a>           |      |       |

At the bottom right, a context menu is open over the Jira Administration row, listing options: 'Manage users', 'Manage product access', 'Open Jira Administration' (which is circled in red), 'Jira settings', and 'Manage Jira apps'.

# Webhook



- Accès à webhook à partir de Jira

The screenshot shows the Jira Settings page. At the top, there is a search bar, a bell icon, a question mark icon, a gear icon (highlighted with a red box), and a CM icon. Below the header, there are two main sections: "ATLASSIAN ADMIN" and "JIRA SETTINGS". In the "ATLASSIAN ADMIN" section, there are links for "User management" and "Billing". In the "JIRA SETTINGS" section, there are links for "System" (highlighted with a red box), "Products", "Projects", "Issues", and "Apps". Each link has a brief description below it.



The screenshot shows the Jira Admin interface. At the top, there is a search bar labeled "Search Jira admin" (highlighted with a red box). Below the search bar are two buttons: "Advanced Settings" and "Edit Settings".

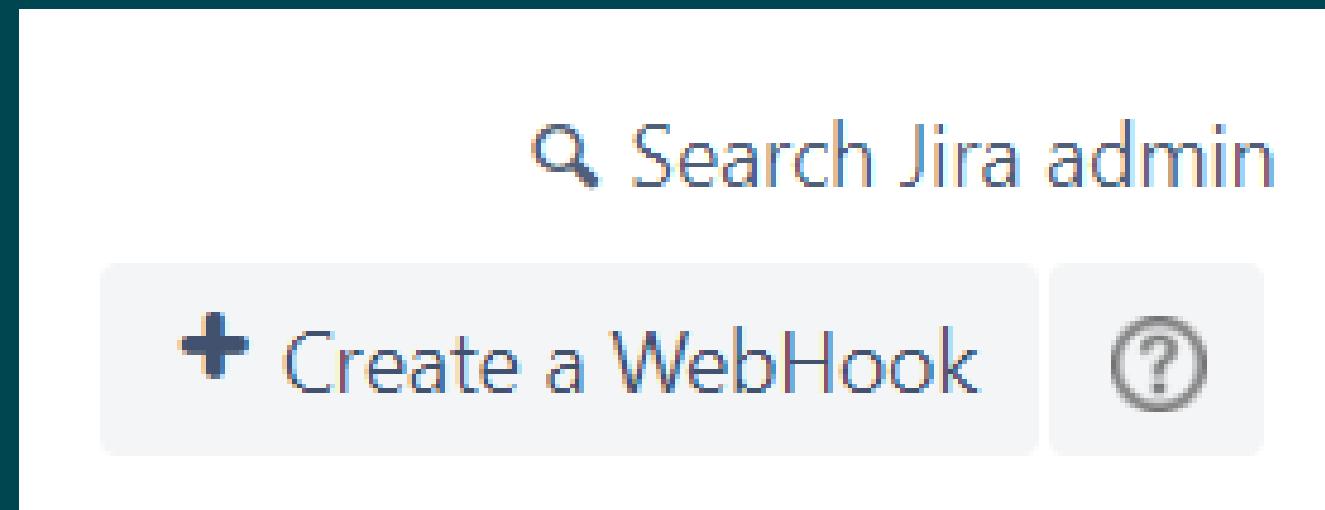


The screenshot shows the Jira Administration screen. At the top, there is a search bar with the text "webhooks" and a close button. Below the search bar, there is a section titled "ADMINISTRATION" with a link "WebHooks" (highlighted with a blue box). At the bottom of the screen, there is a note: "Pressing period (.) opens this dialog box".

# Webhook



- Créer un nouveau webhook



# Webhook



- Url de Postman

The screenshot shows a "Runs" section for a flow named "Running on the cloud". It is currently "WAITING". A red circle highlights the "Webhook URL" field, which contains the value <https://8f7b06d68f7247b0a2d830c71d8c0a48.flows.n8n.net>. Below the field, a note states: "Your Flow is running in the cloud. This webhook URL triggers the Start block; sending a request to the URL will start the flow. You can view a log of the Webhook calls below." At the bottom, it says "Last deployed Today, 11:39 pm".

- Configuration de l'url du webhook

The screenshot shows a configuration form for a "Webhook Listener". The "Name" field is set to "New Webhook Listener". The "Status" field has two options: "Enabled" (which is selected) and "Disabled". The "URL" field is empty. A large red arrow points from the highlighted URL in the left screenshot to this "URL" field.

# Webhook



- Configuration du webhook : Issue

## Issue related events

You can specify a JQL query to send only events triggered by matching issues. The JQL filter does not apply to events under the Issue link and Filter columns.

All issues

[Syntax help](#)

### Issue

- created
- updated
- deleted

### Worklog

- created
- updated
- deleted

### Comment

- created
- updated
- deleted

### Entity property

- created or updated
- deleted

### Attachment

- created
- deleted

### Issue link

- created
- deleted

### Filter

- created
- updated
- deleted

# Webhook



- Configuration du webhook : User

## User related events

### User

- created
- deleted
- updated

- Pour voir les évènements liés aux utilisateurs

# Webhook



- Configuration du webhook : User

## Jira configuration related events

### Features status change (enabled/disabled)

- voting
- watching
- unassigned issues
- subtasks
- attachments
- issue links
- time tracking
- time tracking provider

Voir les :

- subtasks
- attachments
- issue links
- time tracking
- time tracking provider

# Webhook



- Configuration du webhook : Project

## Project related events

Events for projects and project versions.

### Project

- created
- updated
- deleted
- soft\_deleted
- restore\_soft\_deleted
- archived
- restore\_archived

### Version

- released
- unreleased
- created
- moved
- updated
- merged
- deleted

- Rien n'est mit dans les projects : un seul projet

# Webhook



- Configuration du webhook : Jira

## Jira Software related events

Events for sprints and boards

### Sprint

- created
- deleted
- updated
- started
- closed

### Board

- created
- deleted
- updated
- configuration changed

- Pas de sprint
- Pas de board

- Créeer le webhook

Create

Cancel

# Webhook



- Configuration du webhook : Jira

 WebHook 'New Webhook Listener' has been successfully created.

**New Webhook Listener** ENABLED

Last updated: 13/Apr/23 11:56 PM by Mourad Mettioui

URL

<https://8f7b06d68f7247b0a2d830c71d8c0a48.flow.pstmn.io>

# Webhook



- dans issue de Jira

Mourad Mettioui raised this request via Portal  
View request in portal

Description  
i can't connect with postman

Components None

Similar requests

Activity

Show: All Comments History Work log Approvals

Add internal note **Reply to customer** (circled in red)

Pro tip: press M to comment

Normal text B I ... A ...

i will fix that

Save Cancel

Webhook Events  
POST /?triggeredByUser... 00:06:02.205 ✓

Mock Servers Flows

Aa Log

View Live Logs

Online Find and Replace Console

```
{timestamp: 1681423561242, webhookEvent: "comment_created", comment: {...}}
```

All Logs Clear

Forms for Flows! COMING SOON

Create and share forms to collect data and run your deployed flows.

View Live Logs

Online Find and Replace Console

```
{timestamp: 1681423561242, webhookEvent: "comment_created", comment: {...}}  
timestamp: 1681423561242  
webhookEvent: "comment_created"  
comment: {...}  
self: "https://mourad.atlassian.net/rest/api/2/issue/10018/comment/10011"  
id: "10011"  
author: {...}  
body: "i will fix that"  
updateAuthor: {...}  
created: "2023-04-14T00:06:01.242+0200"  
updated: "2023-04-14T00:06:01.242+0200"  
jsdPublic: true
```

# Méthode RUP

## Rational Unified Process

Se décompose en 4 grandes phases :

- L'inception : recueillir les exigences et enjeux
- La conception : spécifier l'objet du projet (+ description fonctionnelle et technique)
- La réalisation : développer puis intégrer tous les éléments nécessaire au projet
- La transition : transformer le projet en produit



BY-SYSTEMS

**ECAM**

BRUSSELS ENGINEERING SCHOOL

# Présentation WEEK 3

Van Offen Victor  
Maillard Corentin  
Mettioui Mourad

