

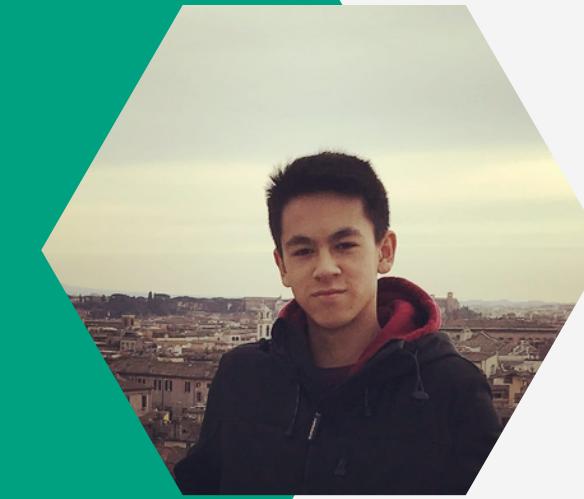
Presentation

WEEK 5

Van Olffen Victor
Maillard Corentin
Mettioui Mourad

Our team

Do not hesitate to contact us !



Van Olffen Victor
20253@ecam.be



Maillard Corentin
21306@ecam.be



Mettioui Mourad
195019@ecam.be

Content

- ETL Diagram
- Jira-Connector
- Back-End
- Front-End

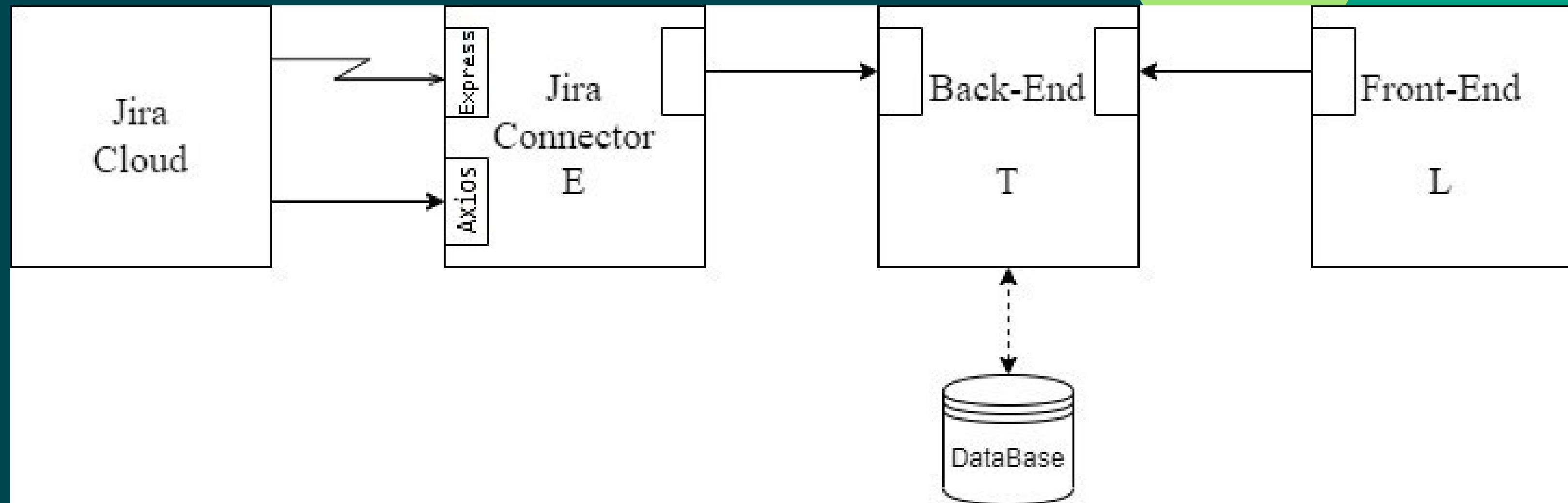
With Jira and VisualStudioCode

Content

- ETL Diagram
- Jira-Connector
- Back-End
- Front-End

With Jira and VisualStudioCode

ETL Diagram



- E: Extract

- T: Transform

- L: Load

EXPLORER

ECAM2023

- > .vscode
- > ecam2023-stage
- ✓ jira-connector
 - > build
 - > log
 - > node_modules
- ✓ src
 - ✓ client
 - TS httprequest.ts M
 - > lib
 - > server
- .eslintrc.json
- .gitignore M
- .prettierrc.json
- package-lock.json M
- package.json M
- README.md M
- tsconfig.json
- > log
- > nodejs-express-sequelize-pos...
- > sipsecure_notify_react-master
- > sphinx-doc
- > ts_prj_template
- > ts-project

TS httprequest.ts M X

```
jira-connector > src > client > TS httprequest.ts > ⚡ <function>
197
198   displayName:
199     responseOrganizationCustomer.data.values[j].displayName,
200   );
201   axios.post('http://10.1.100.244:8080/api/v1/jira/customer/', {
202     accountId: responseOrganizationCustomer.data.values[j].accountId,
203     accountType: 'customer',
204     emailAddress:
205       responseOrganizationCustomer.data.values[j].emailAddress,
206     displayName:
207       responseOrganizationCustomer.data.values[j].displayName,
208   );
209   }
210   customers.push(customer);
211
212   // customerDB
213   //regarde si organisation déjà présentent dans la database
214   const foundView = viewDB.find(obj => {
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL GITLENS

bash + ⌂ ⌂ ... ⌂ ⌂ ⌂

```
coren@Renco74 MINGW64 /d/document/ECAM2023/jira-connector (main)
$ npm run nodehttp --clientid=nr3DZPJ129vipNSAwMPBiAk0AZDBIs0I --clientsecret=ATOAa7cwLZ80GoDvHTBInIjh-IJgKrndrYA7ubmF-Km46t
b1QiNKxJGBwv3SvFFoL4HJC8918369
```

Admin 4

Object Tools Help

Servers

Servers (1)

PostgreSQL 15

Databases (2)

node express-sequelize

- Casts
- Catalogs
- Event Triggers
- Extensions
- Foreign Data Wrappers
- Languages
- Publications

Schemas (1)

public

- Aggregates
- Collations
- Domains
- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences

Tables (6)

- jira_customers
- jira_organizations
- jira_users
- tutorials
- users_jira
- view_customer_organizations

Trigger Functions

Types

Views

Subscriptions

postgres

Login/Group Roles

Tablespaces

Dashboard Properties SQL public.jira.orga... public.jira.customers/node-express-sequelize/postgres@PostgreSQL 15

public.jira.customers/node-express-sequelize/postgres@Postg...

No limit

Query History

Scratch Pad

Query

```
1 SELECT * FROM public.jira_customers
2 ORDER BY id ASC
```

Data Output Messages Notifications

#	ID	AccountId	AccountType	EmailAddress	DisplayName	createdAt	updatedAt
1	[PK] integer	character varying (255)	character varying (255)	character varying (255)	character varying (255)	timestamp with time zone	timestamp with time zone

Total rows: 0 of 0 Query complete 00:00:00.470

Ln 2, C

Search by issue_id

Search

Issues per page: 5 ▾

Issues List

1

Summary	Description	Key	Time spent	Updated	Status	Organization id	User id	Actions
Présentation week 4	Présentation week 4	ECAM-7	18000	2023-05-04T17:15:50.359+0200	Open	1	70121:a1214ae7-5c7a-422f-8513-edbeb36d6838	<button>Edit</button> <button>View worklogs</button>
Présentation week 3	Jira webhook avec typescript	ECAM-4	72000	2023-05-04T16:27:12.193+0200	Open	1	70121:a1214ae7-5c7a-422f-8513-edbeb36d6838	<button>Edit</button> <button>View worklogs</button>
Test	voir si je peux m'auto envoyer	ECAM-2	57600	2023-04-19T14:51:42.523+0200	Waiting for support	1	70121:a1214ae7-5c7a-422f-8513-edbeb36d6838	<button>Edit</button> <button>View worklogs</button>
noob	big noob	ECAM-1		2023-05-03T09:49:10.742+0200	Waiting for customer	2	qm:2254bb2d-80ac-40c0-b4e8-4ee36123bb1f:7f6f845d-14b0-4b78-8da6-92205f70ea28	<button>Edit</button> <button>View worklogs</button>

Content

- ETL Diagram
- Jira-Connector
- Back-End
- Front-End

With Jira and VisualStudioCode

Jira-Connector

- Utilities
- Http requests
- Webhooks

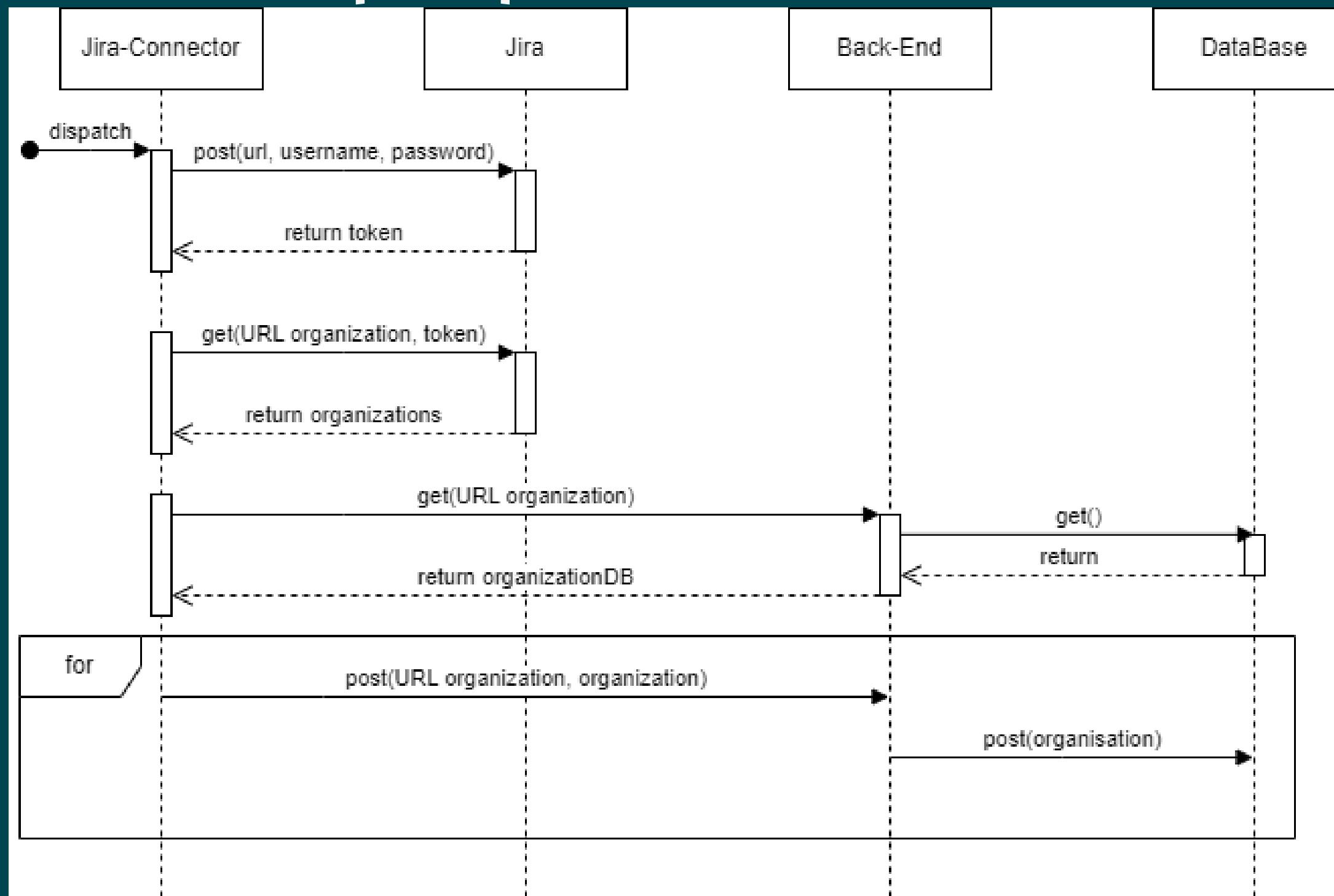
Jira-Connector

Utilities

- Link between Jira Service Management - Back-End
- Send filtered and formatted data to the Back-End
- Webhook event (CRUD) send to the Back-End
- Keep all message from http request and webhook into a JSON file.
 - Format for a message:
{TIME STAMP, SEVERITY, AXIOS/EXPRESS, CMD, RESPONSE}

Jira-Connector

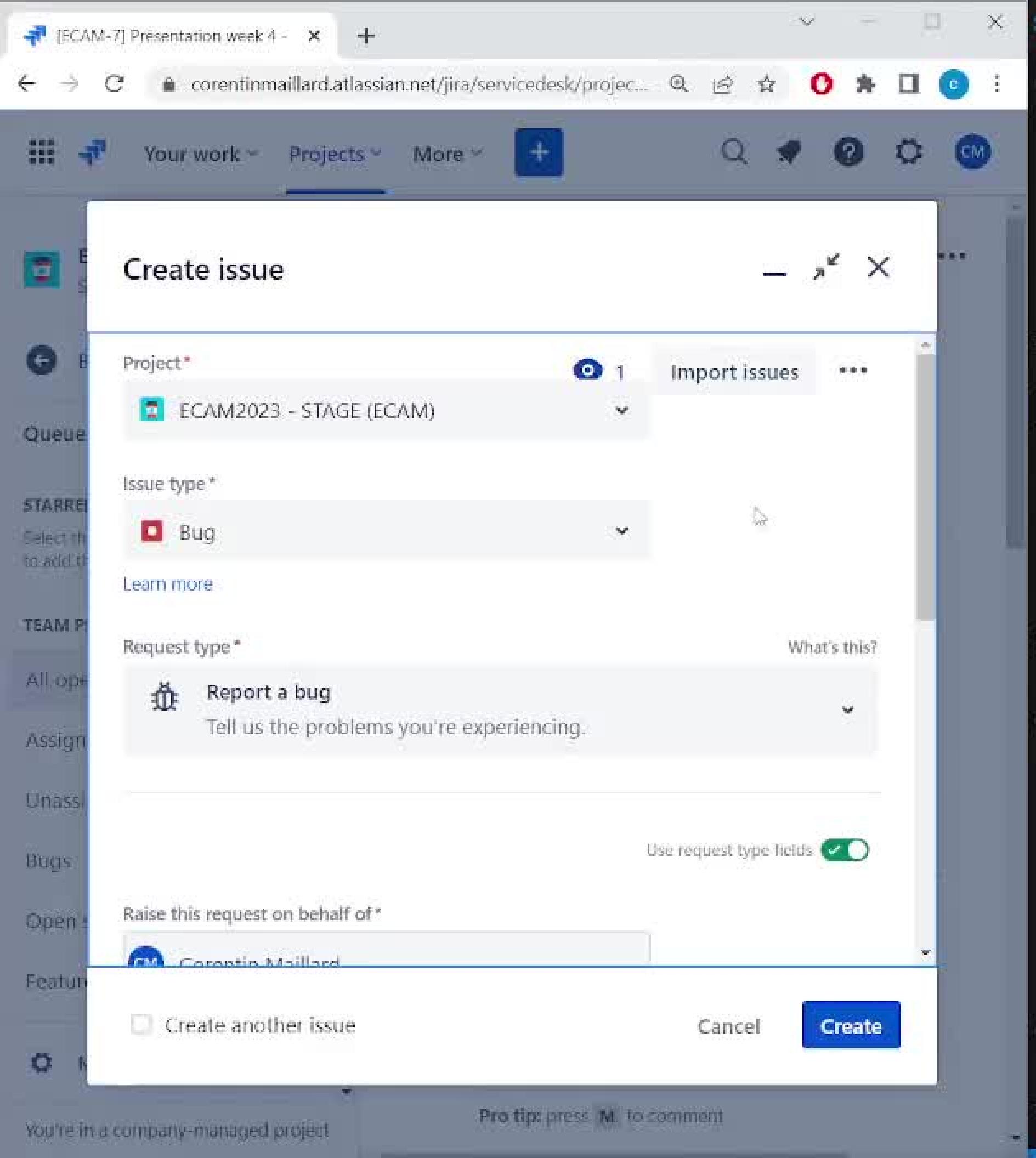
UML Organizations : Http requests with API token



Jira-Connector

Http requests :

- Utilisation of :
 - API token
 - Promises (async - await)
 - Axios (request POST, GET)
- Retrieve from Jira :
 - Users and customers
 - Organizations
 - Issues and associated worklogs



jira_http.json - ECAM2023 - Visual Studio Code

jira_connector > log > {} jira_http.json

1

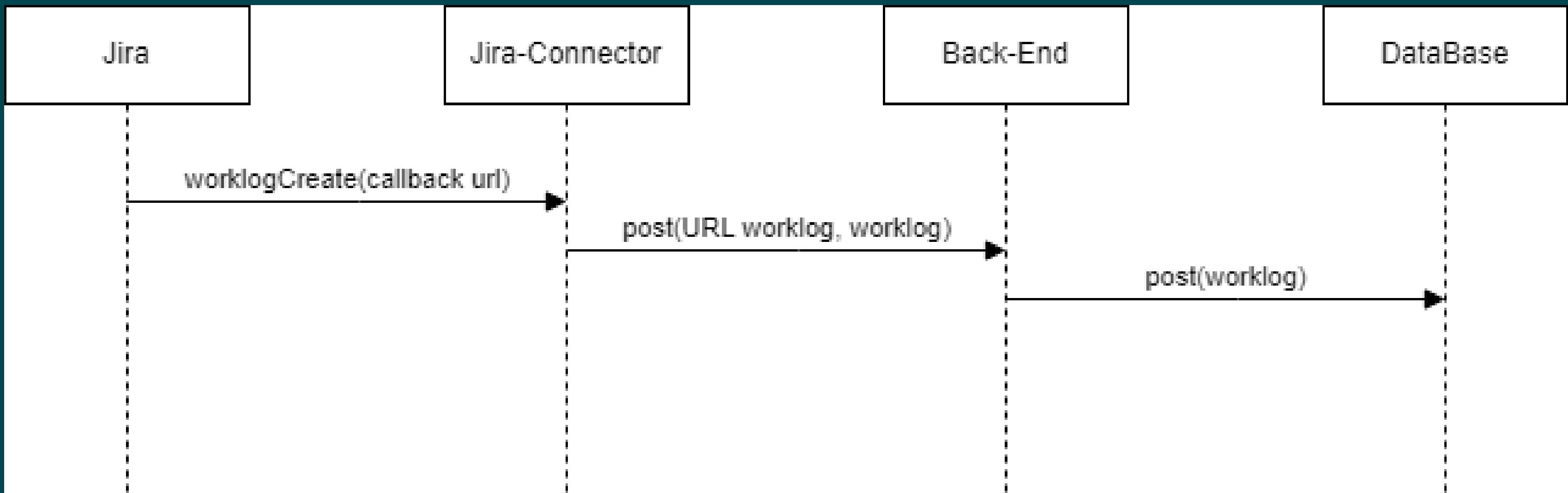
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL ...

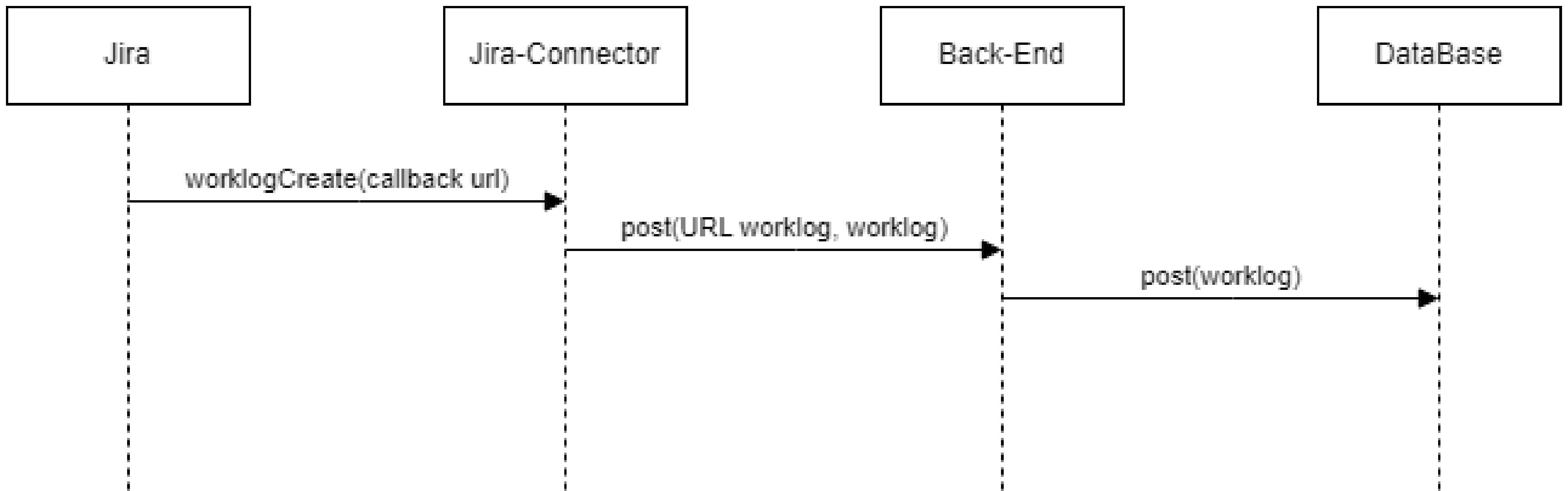
> tsc -p .

```
coren@Renco74 MINGW64 /d/document/ECAM2023/jira-connector (main)
$ npm run nodehttp --atlassianEmail=corentin.maillard@gmail.com --atlassianToken=ATATT3xFfGF0UAfDpmqe4U6WHMfgpuYsd0hwnkn2phfE1IS2syeG9xaMSIjiTQAxUY5cRtzgeWN
SGev2U_pJ01EsaebiY-_vPF94_4EsLVOjoW6AkMYZvAzmfSv2IY2FjfmX6gy13gvkVysTeRQZQzFlh
oX2rOeBbxwkleRggP_7ucVtSWjz0g4=569F7D18 --baseUrl=https://corentinmaillard.atl
assian.net/
```

Jira-Connector

UML Worklog : Webhooks





Jira-Connector

Webhooks :

- Configuration of the webhook event on Jira (separated)
- Utilisation of :
 - Ngrok (Callback URL send on port 3000)
 - Express
- Listened event :
 - Issues (CRUD)
 - Worklog (CRUD)
- Send request (POST, GET, PUT) to the back-end using axios

Content

- ETL Diagram
- Jira-Connector
- Back-End
- Front-End

featuring Jira and VisualStudioCode

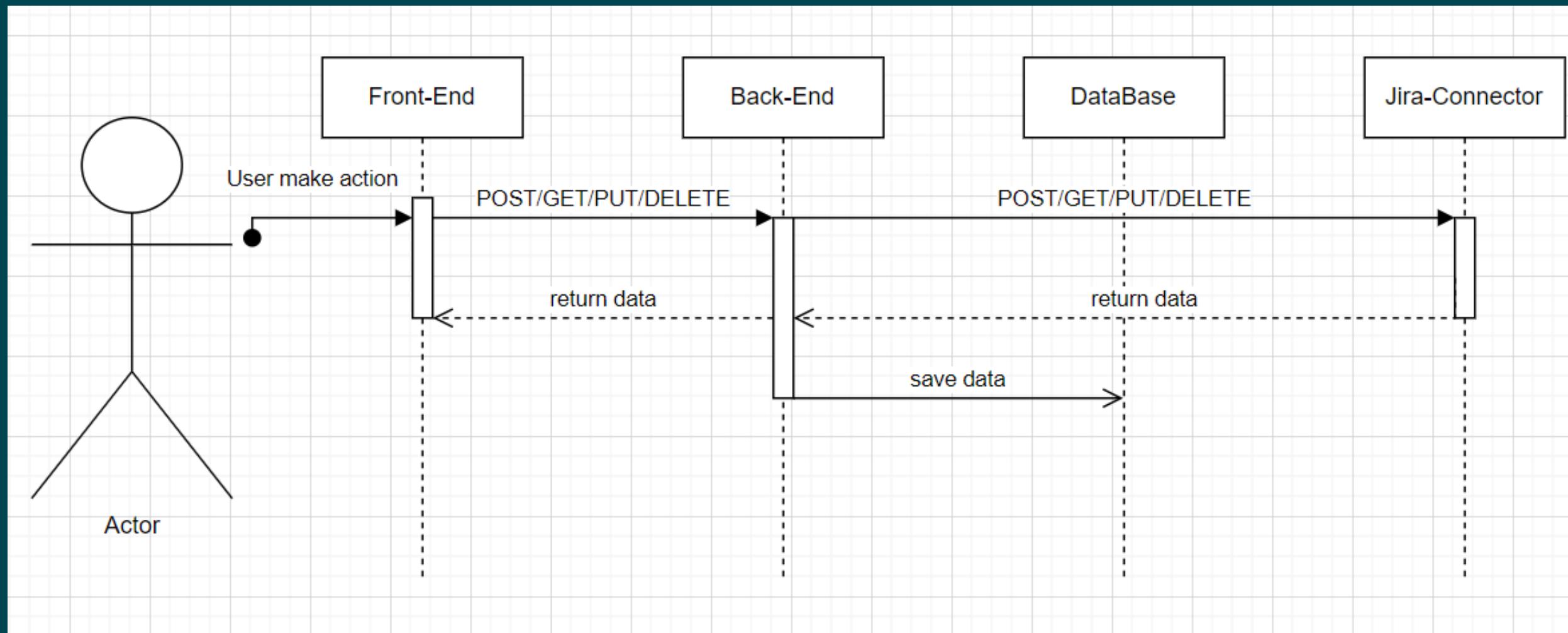
Back-End



- Definition
- Express
- axios vs Fetch

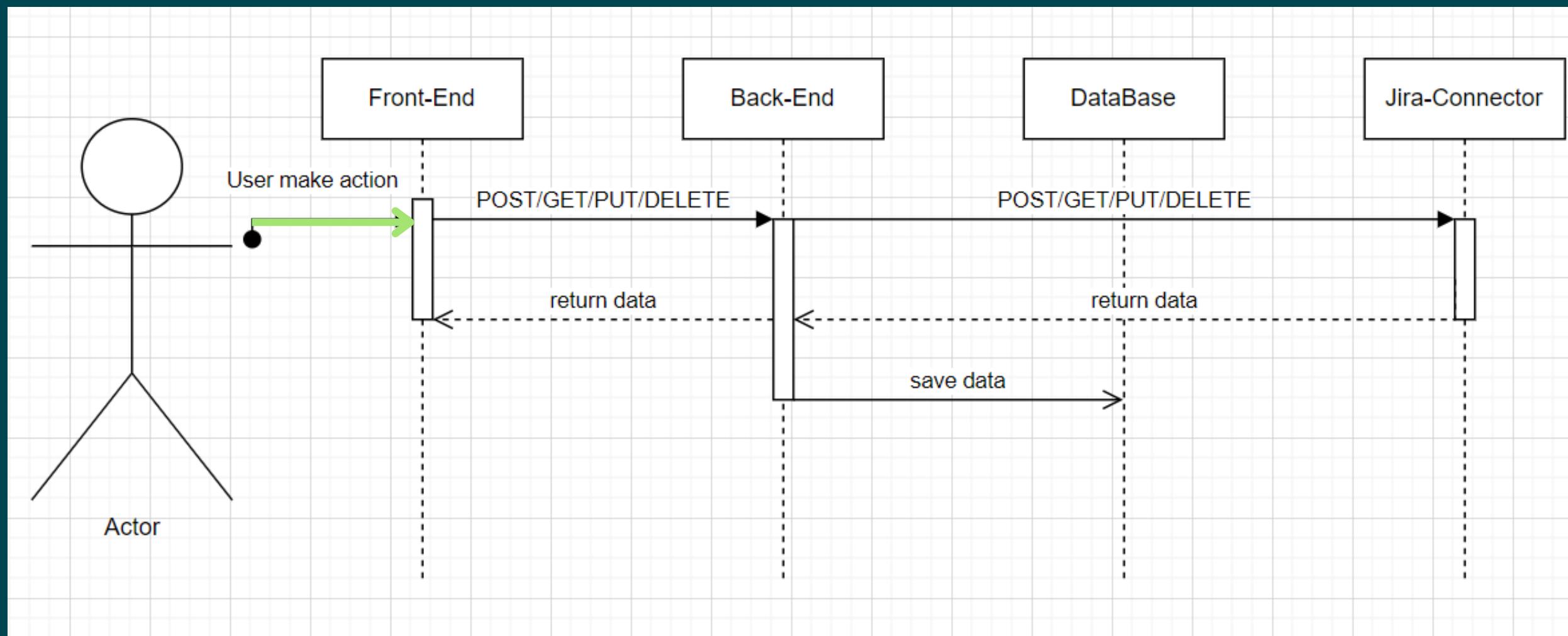
Back-End

- Definition



Back-End

- Definition



Back-End

- Definition

Jira Connector Issues Users Organizations Customers Add new ▾

Search by organizationID Search

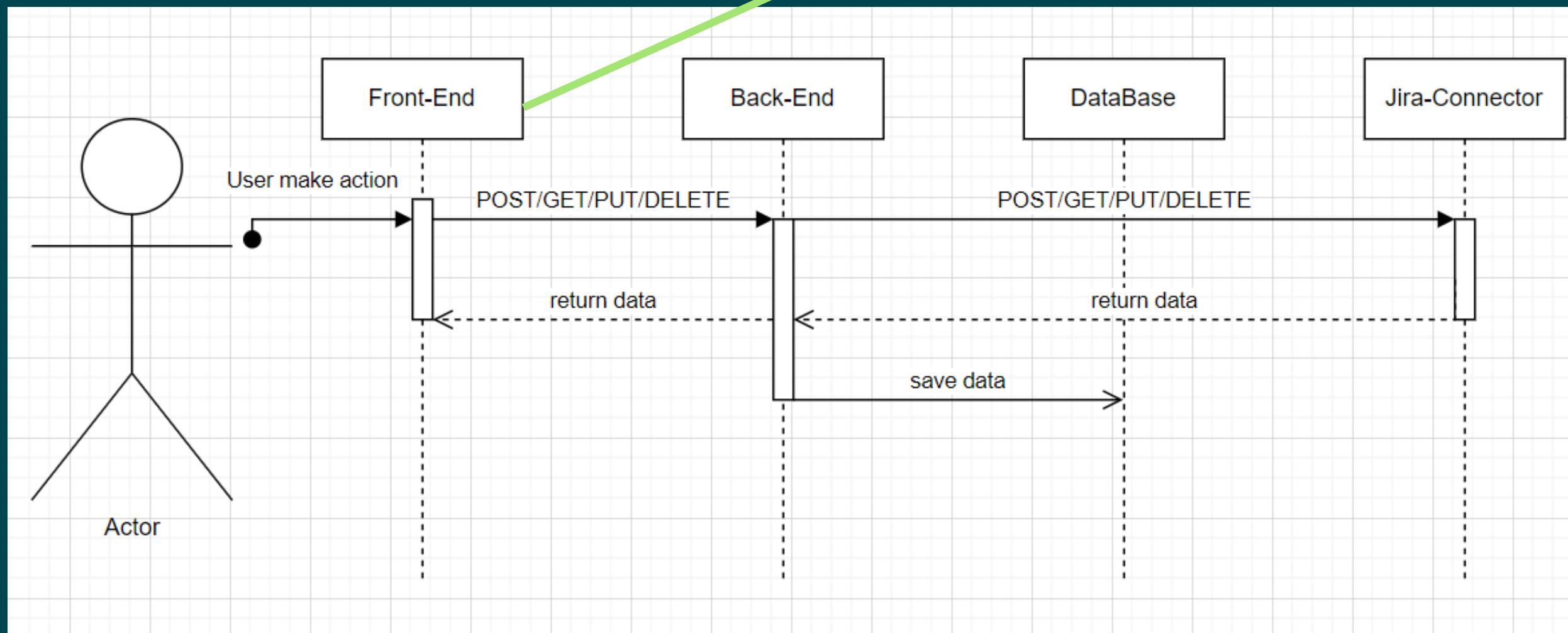
Organizations per page : 5

Organizations List

Name	id	Actions
ECAM	1	Edit View customers
BySystems	2	Edit View customers

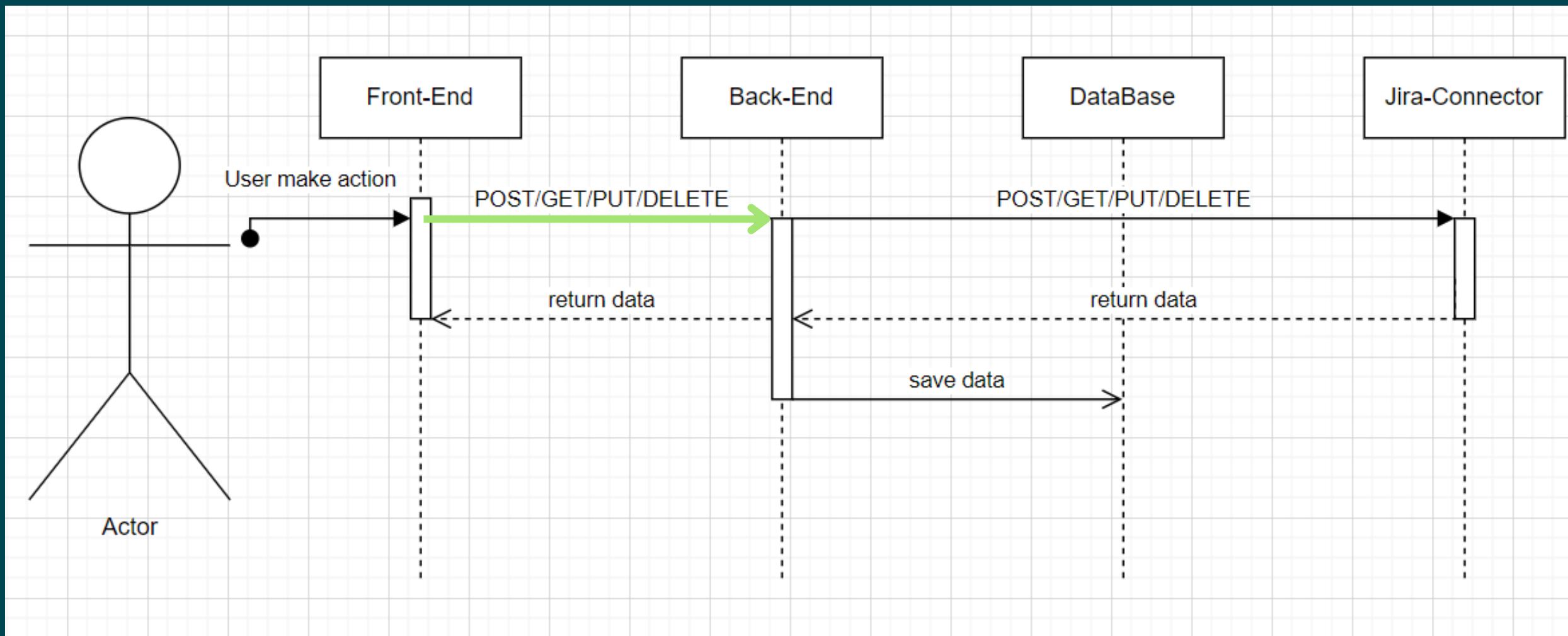
Export organizations list as : [xlsx](#) [Exporter en xlsx](#)

ECAM © 2023. All rights reserved.



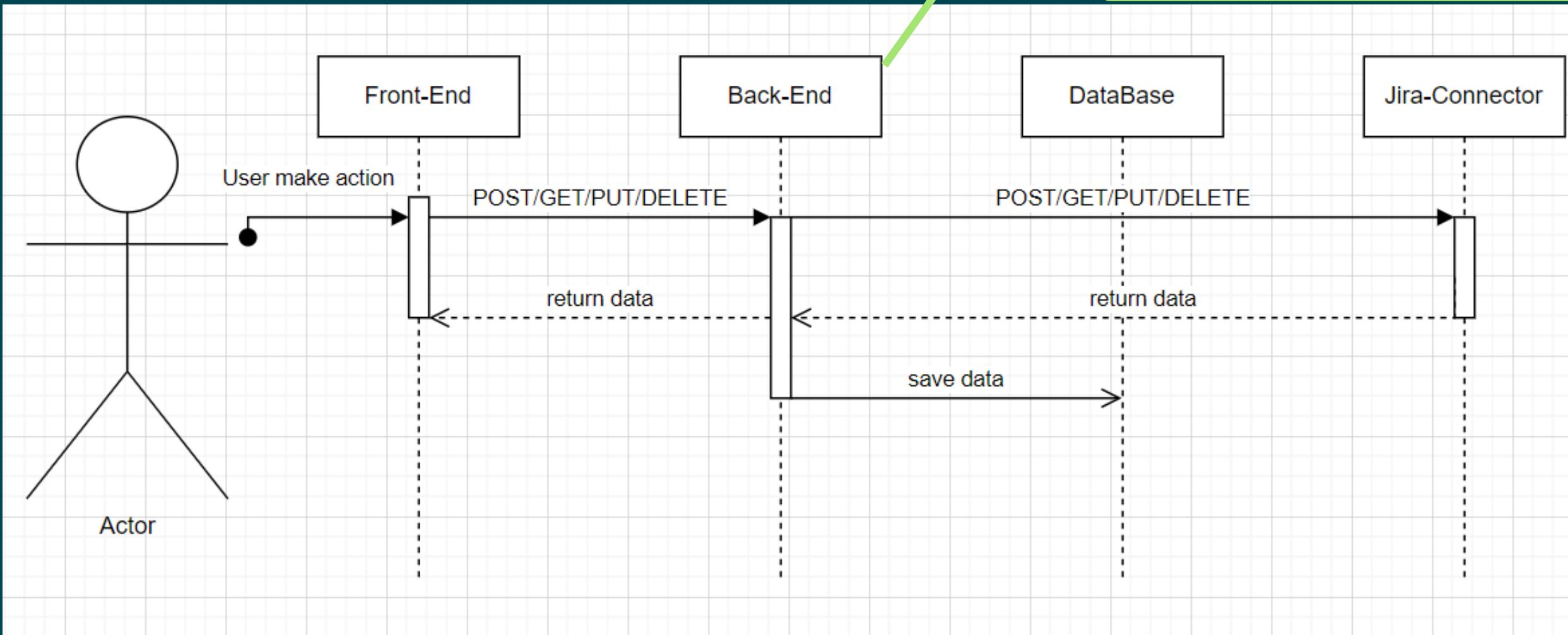
Back-End

- Definition



Back-End

- Definition



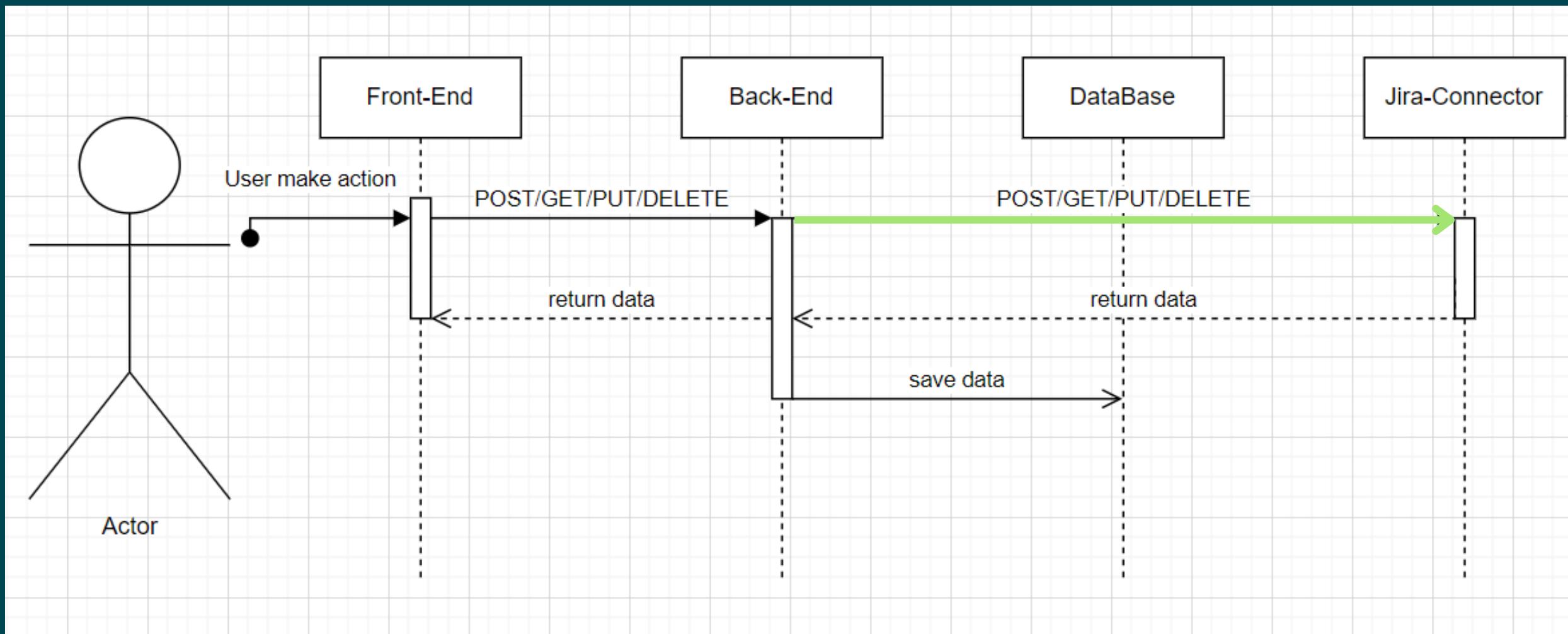
```
33 // Routes - Jira
34 require("./app/routes/jira_user.routes")(app);
35
36
37 // set port, listen for requests
38 const PORT = process.env.PORT || 8080;
39 app.listen(PORT, () => {
40   console.log(`Server is running on port ${PORT}.`);
41 });
42
```

TERMINAL OUTPUT DEBUG CONSOLE

```
_get_indexdef(ix.indexrelid) AS definition FROM pg_class t, pg_class i, pg_index ix, pg_attribute a WHERE t.oid = ix.relnum
relname = 'jira_users' GROUP BY i.relnum, ix.indexrelid, ix.indisprimary, ix.indisunique, ix.indkey ORDER BY i.relnum
PS C:\Program Files\jira-connector\Node-Express-Sequelize-PostgreSQL> node server.js
Server is running on port 8080.
Executing (default): CREATE TABLE IF NOT EXISTS "tutorials" ("id" SERIAL , "title" VARCHAR(255), "description" VARCHAR(255),
"createdAt" TIMESTAMP WITH TIME ZONE NOT NULL, PRIMARY KEY ("id"));
Executing (default): SELECT i.relnum AS name, ix.indisprimary AS primary, ix.indisunique AS unique, ix.indkey AS indiskey
FROM pg_class t, pg_class i, pg_index ix, pg_attribute a WHERE t.oid = ix.relnum
relname = 'tutorials' GROUP BY i.relnum, ix.indexrelid, ix.indisprimary, ix.indisunique, ix.indkey ORDER BY i.relnum
Executing (default): CREATE TABLE IF NOT EXISTS "jira_users" ("id" SERIAL , "accountid" VARCHAR(255), "accounttype"
TIMESTAMP WITH TIME ZONE NOT NULL, "updatedAt" TIMESTAMP WITH TIME ZONE NOT NULL, PRIMARY KEY ("id"));
Executing (default): SELECT i.relnum AS name, ix.indisprimary AS primary, ix.indisunique AS unique, ix.indkey AS indiskey
FROM pg_class t, pg_class i, pg_index ix, pg_attribute a WHERE t.oid = ix.relnum
relname = 'jira_users' GROUP BY i.relnum, ix.indexrelid, ix.indisprimary, ix.indisunique, ix.indkey ORDER BY i.relnum
```

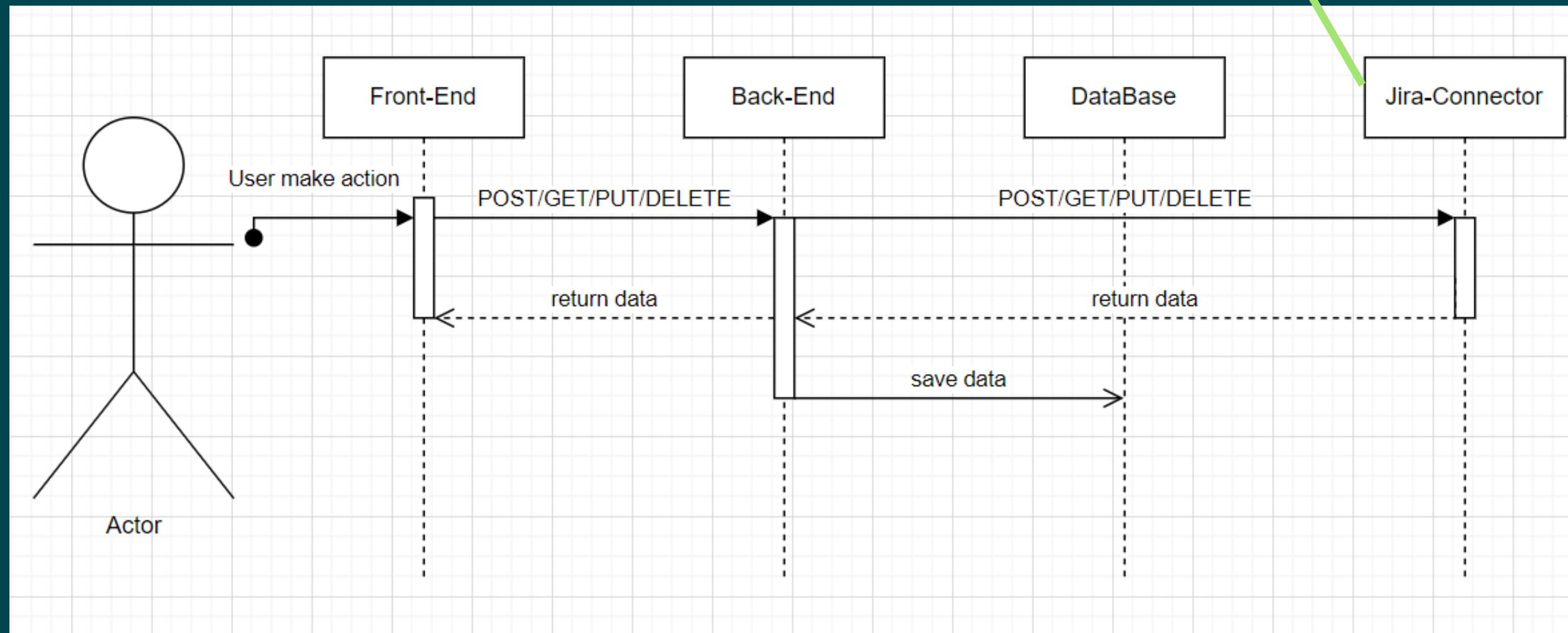
Back-End

- Definition



Back-End

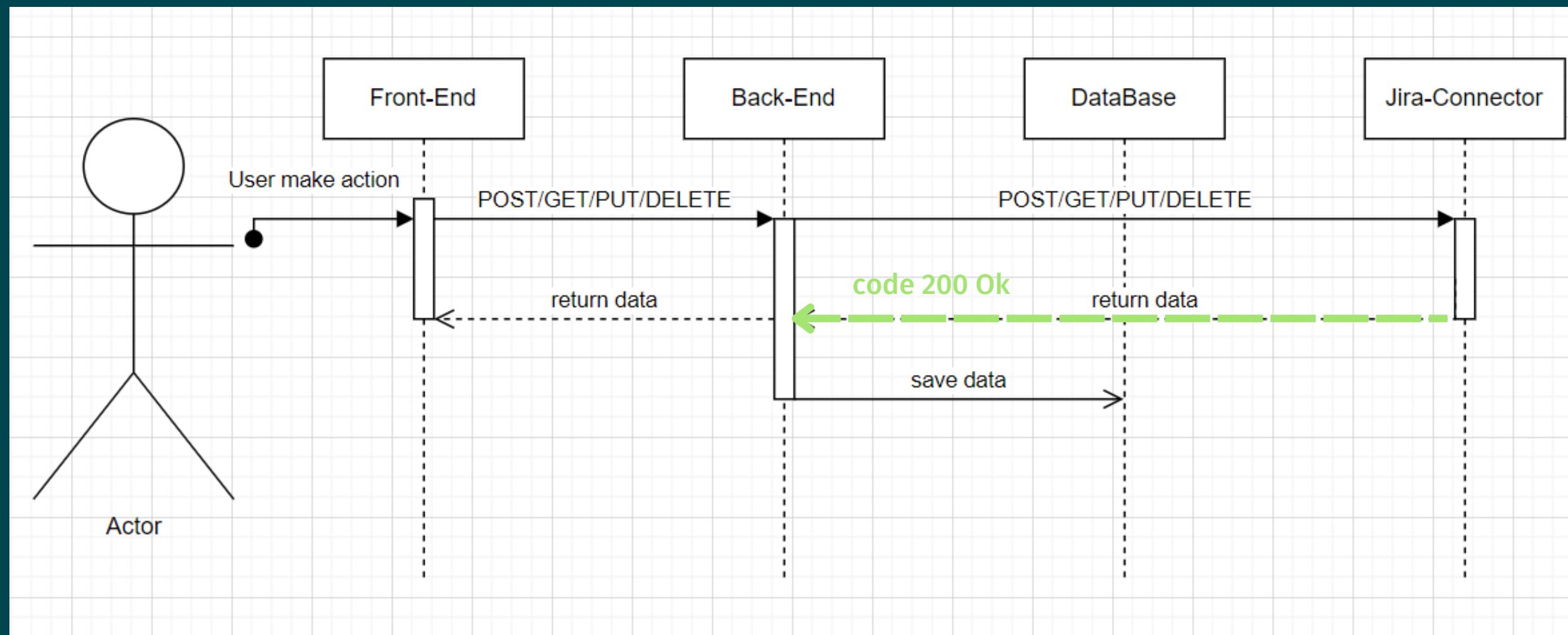
- Definition



```
TERMINAL OUTPUT DEBUG CONSOLE
port: 8080
}
}
PS C:\Progra Stage\jira-connector1\ts_prj_template> npm run node
> projet-typescript@1.4.0 node
> node ./build/server/webhook.js
Serveur démarré sur le port 3000
{"level":"info","message":"Issue updated : 10030"}
{"level":"info","message":"Issue updated : 10030"}
{"level":"info","message":"Comment created : now you know how to fix this"}
Terminer le programme de commandes (O/N) ? o
PS C:\Progra Stage\jira-connector1\ts_prj_template> npm run node
> projet-typescript@1.4.0 node
> node ./build/server/webhook.js
Serveur démarré sur le port 3000
{"level":"info","message":"Comment created : how ?"}
{"level":"info","message":"Worklog created : 18000"}
```

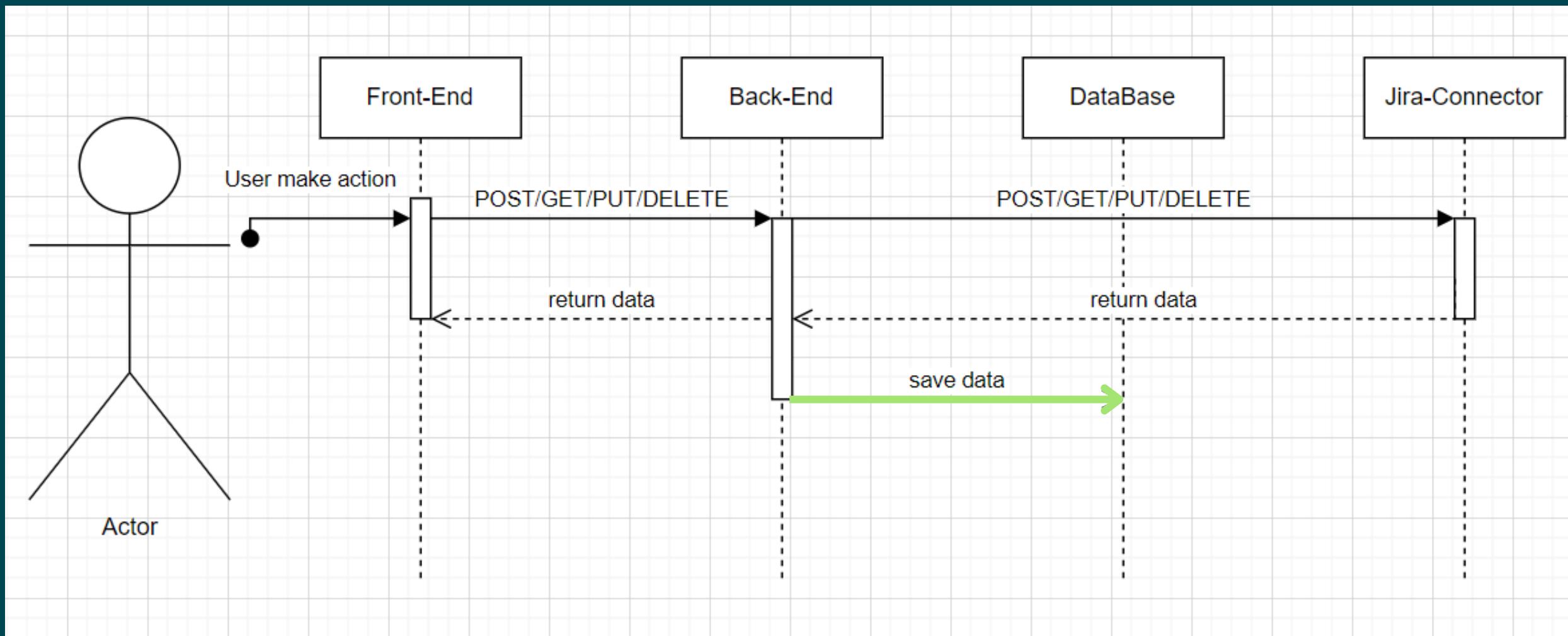
Back-End

- Definition



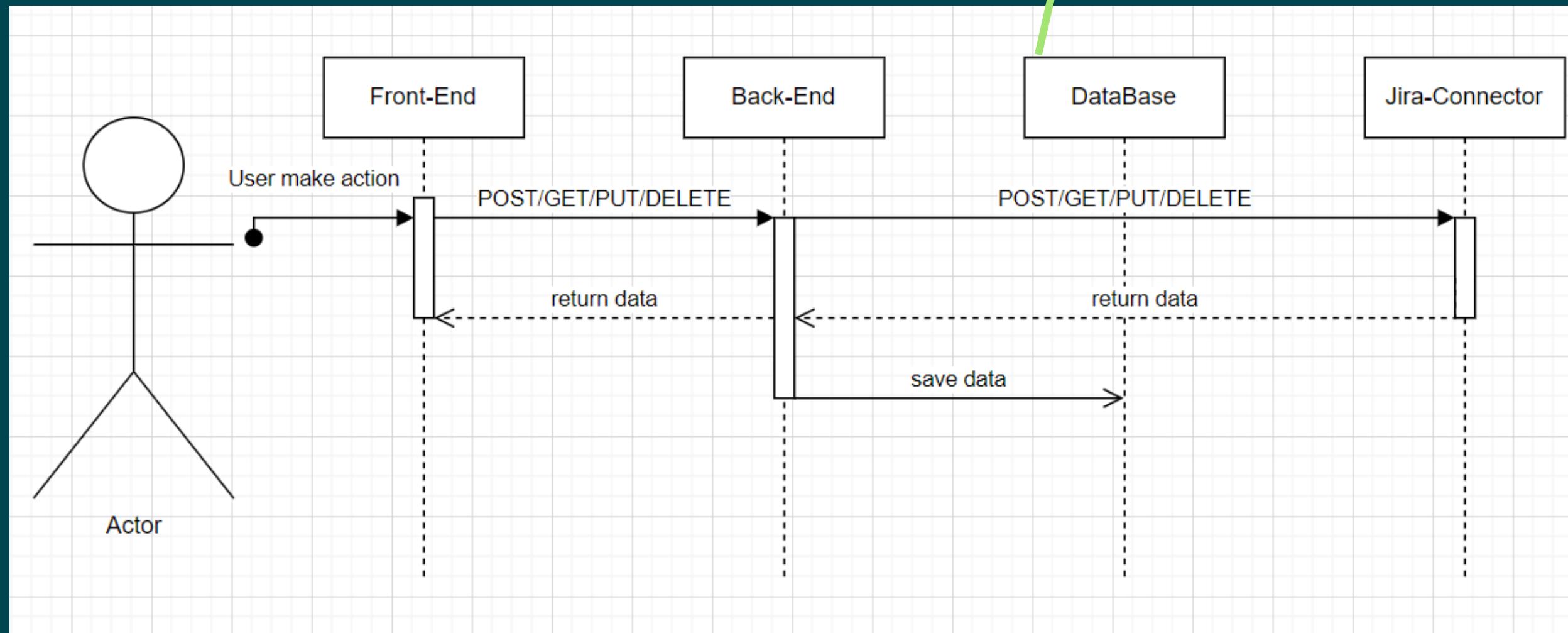
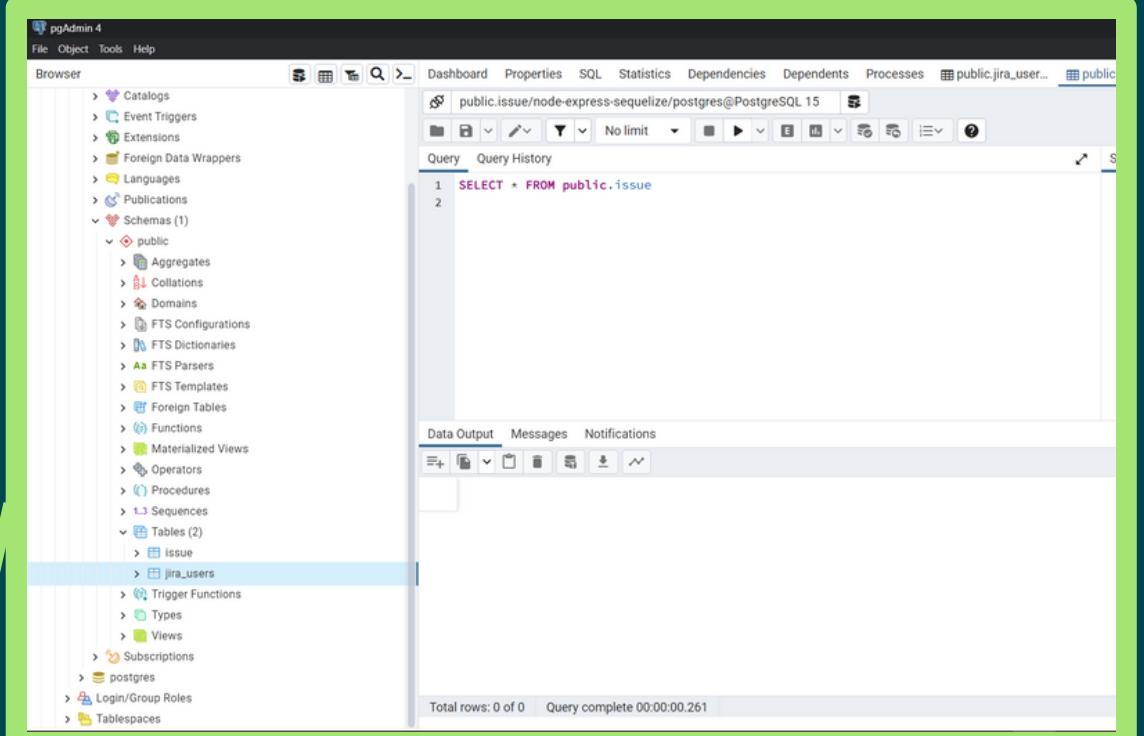
Back-End

- Definition



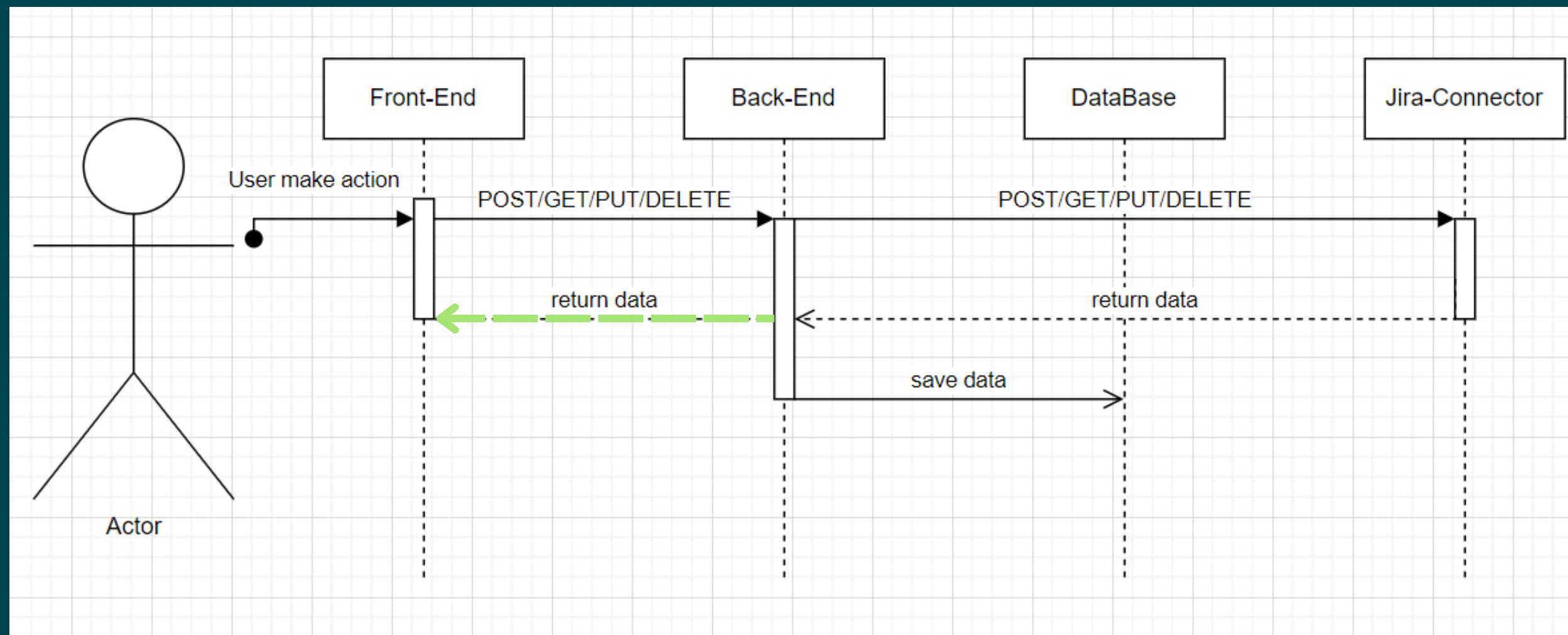
Back-End

- Definition



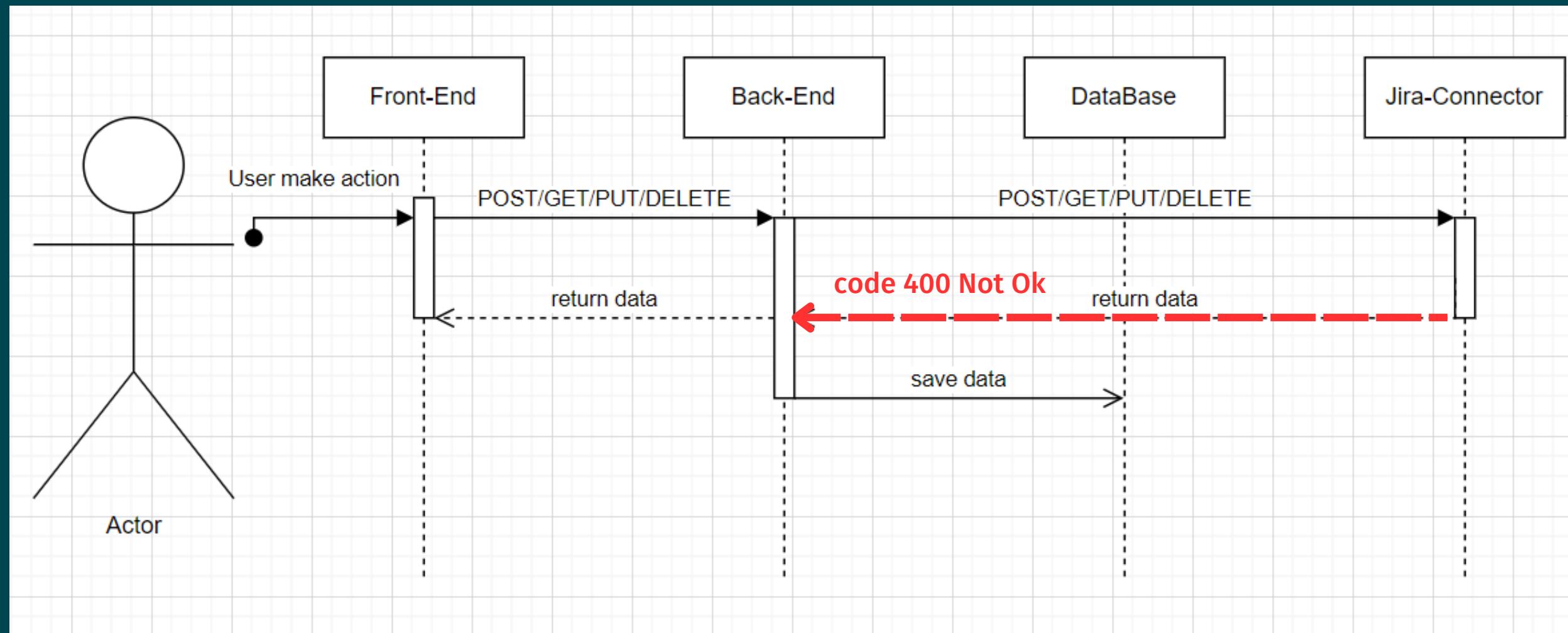
Back-End

- Definition



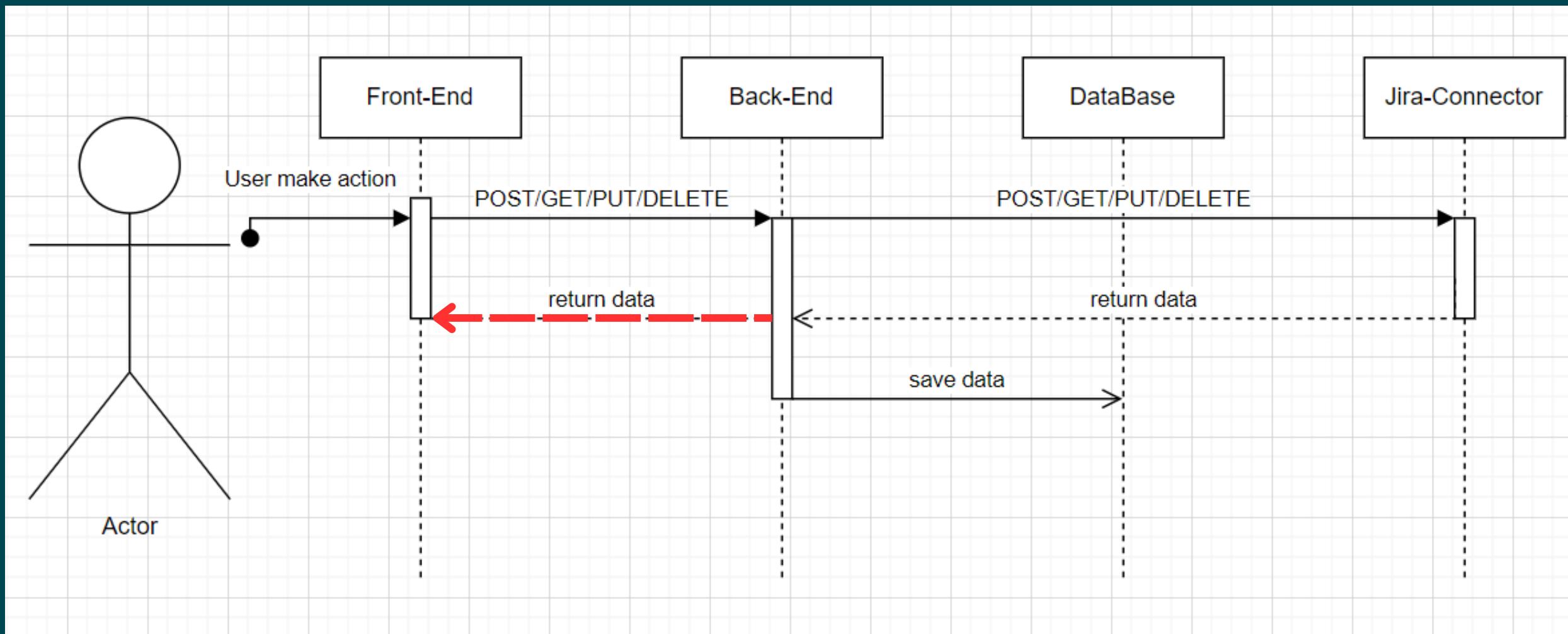
Back-End

- Definition



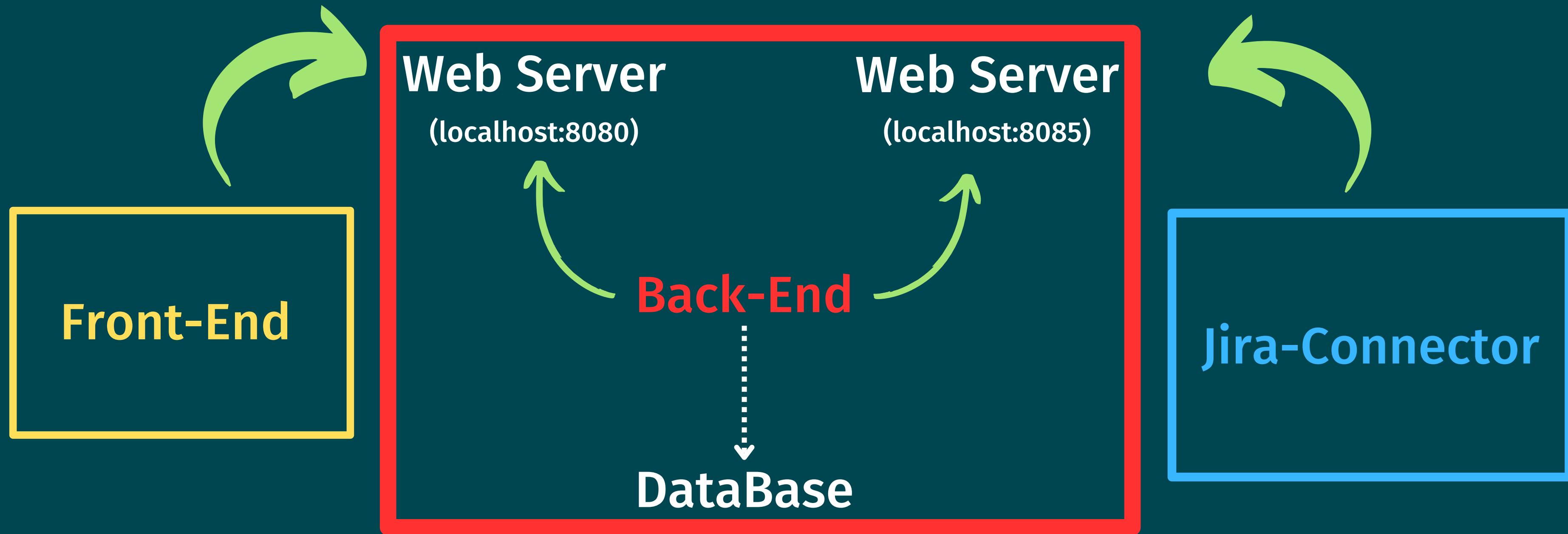
Back-End

- Definition



Back-End

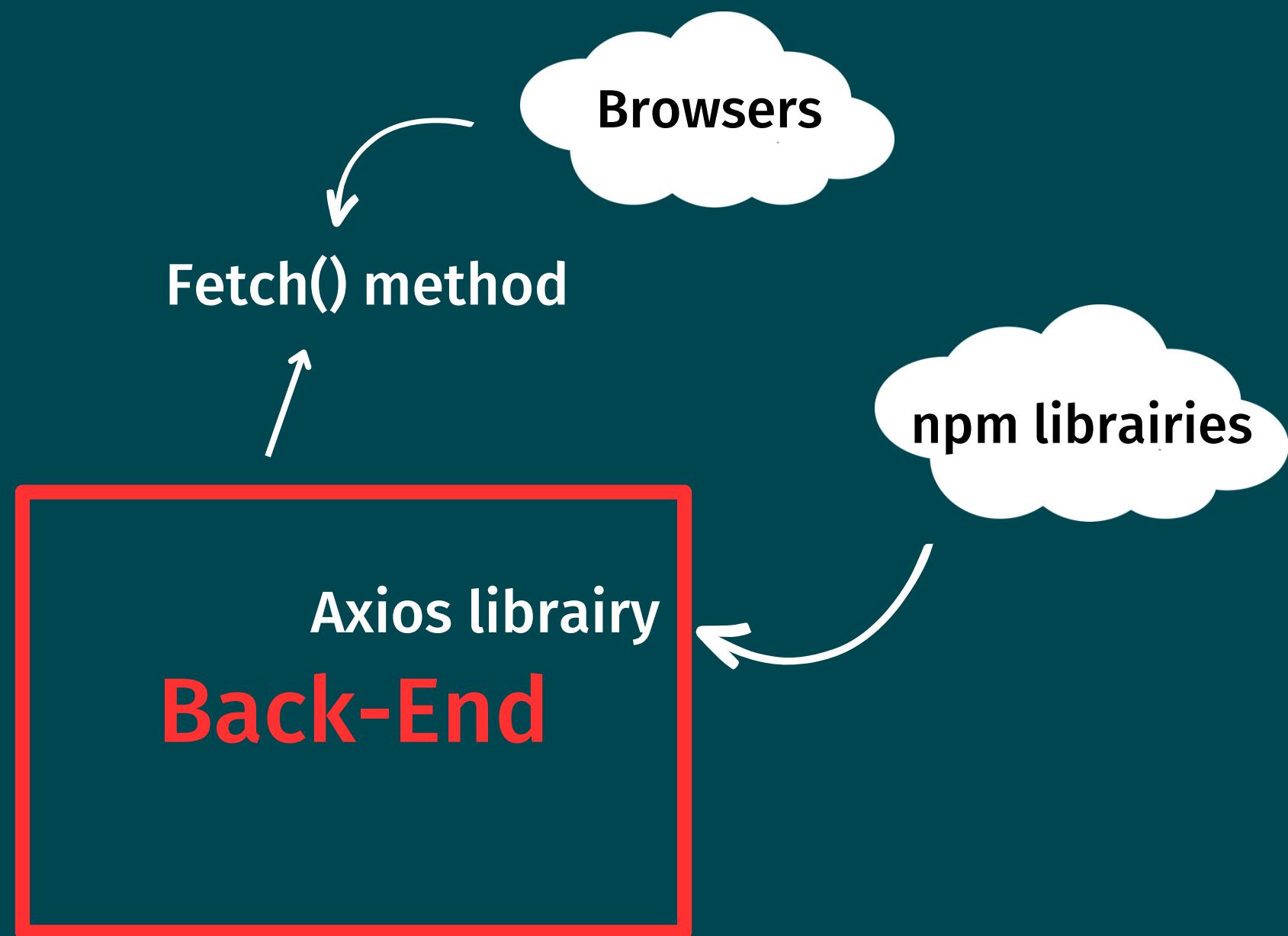
- Express



Axios vs Fetch

- Axios is a javascript library (backend and frontend)
- Fetch is an API Method built into browsers (frontend only)

Axios vs Fetch



Back-End

- issue by organisation

Data Output													Messages	Notifications	
	id [PK] integer	issue_id character varying (255)	key character varying (255)	name issueType character varying (255)	timespent character varying (255)	updated character varying (255)	description character varying (255)	status character varying (255)	summary character varying (255)	userId character varying (255)	organizationId character varying (255)		organizationid character varying (255)		
1		1 123								123					
2		2 10018	ECAM-8	Bug	[null]	2023-05-05T13:56:07.634+0200	Example new issue when Jira connector offline	Open	Presentation week 5	70121:a1214ae7-5c7a-422f-8513-edbeb36d6838			1		
3		3 10017	ECAM-7	Bug	19800	2023-05-05T14:00:43.419+0200	Présentation week 4	Open	Présentation week 4	70121:a1214ae7-5c7a-422f-8513-edbeb36d6838			1		
4		4 10014	ECAM-4	Bug	72000	2023-05-04T16:27:12.193+0200	Jira webhook avec typescript	Open	Présentation week 3	70121:a1214ae7-5c7a-422f-8513-edbeb36d6838			1		
5		5 10012	ECAM-2	Support	57600	2023-04-19T14:51:42.523+0200	voir si je peux m'auto envoyer	Waiting for support	Test	70121:a1214ae7-5c7a-422f-8513-edbeb36d6838			1		
6		6 10011	ECAM-1	Support	[null]	2023-05-03T09:49:10.742+0200	big noob	Waiting for customer	noob	qm:2254bb2d-80ac-40c0-b4e8-4ee36123bb1f:7f6f845d-14b0-4b78-8da6-92205f70ea...	2				

Back-End

- issue by organisation

node-express-sequelize / issues / get issue

GET localhost:8080/api/v1/jira/organization/1/issue

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 [ {  
2   "id": 2,  
3   "issue_id": "10018",  
4   "key": "ECAM-8",  
5   "nameIssueType": "Bug",  
6   "timespent": null,  
7   "updated": "2023-05-05T13:56:07.634+0200",  
8   "description": "Example new issue when Jira connector offline",  
9   "status": "Open",  
10  "summary": "Presentation week 5",  
11  "userId": "70121:a1214ae7-5c7a-422f-8513-edbeb36d6838",  
12  "organizationid": "1",  
13  "createdAt": "2023-05-05T12:48:51.549Z",  
14  "updatedAt": "2023-05-05T12:48:51.549Z"  
15 },  
16 {  
17   "id": 3,  
18   "issue_id": "10017",  
19   "key": "ECAM-7",  
20   "nameIssueType": "Bug",  
21   "timespent": "49800",  
22   "updated": "2023-05-05T14:00:43.419+0200",  
23   "description": "Présentation week 4",  
24   "status": "Open",  
25   "summary": "Présentation week 4",  
26   "userId": "70121:a1214ae7-5c7a-422f-8513-edbeb36d6838",  
27   "organizationid": "1",  
28   "createdAt": "2023-05-05T12:48:51.882Z",  
29   "updatedAt": "2023-05-05T12:48:51.882Z"  
30 },  
31 ],  
32 {  
33   "id": 4,  
34   "issue_id": "10014",  
35   "key": "ECAM-4",  
36   "nameIssueType": "Bug",  
37   "timespent": "72000",  
38   "updated": "2023-05-04T16:27:12.193+0200",  
39   "description": "Jira webhook avec typescript",  
40   "status": "Open",  
41   "summary": "Présentation week 3",  
42   "userId": "70121:a1214ae7-5c7a-422f-8513-edbeb36d6838",  
43   "organizationid": "1"  
44 }
```

node-express-sequelize / issues / get issue

GET localhost:8080/api/v1/jira/organization/2/issue

Params Authorization Headers (7) Body Pre-request Script Tests Settings

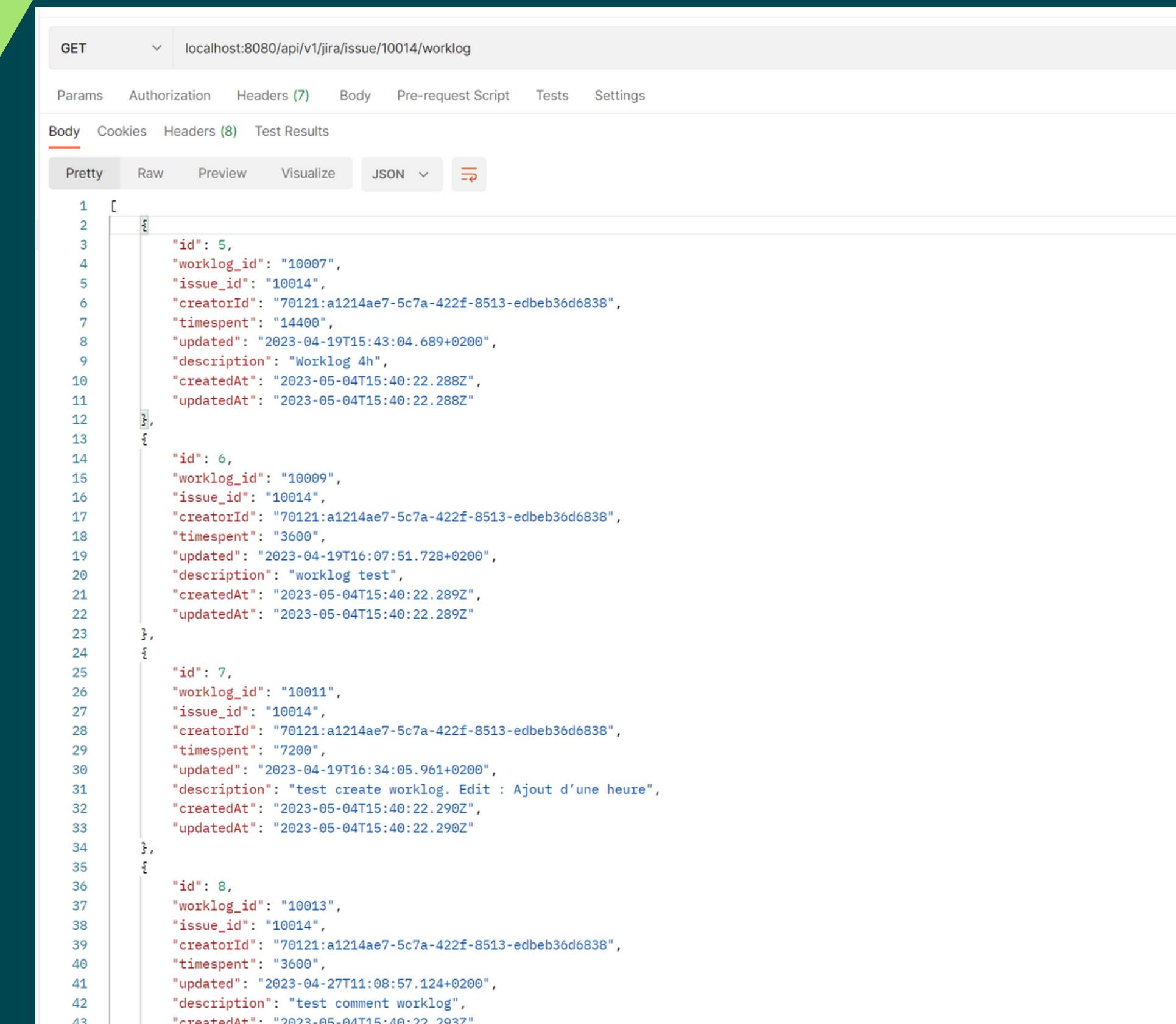
Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1 [ {  
2   "id": 6,  
3   "issue_id": "10011",  
4   "key": "ECAM-1",  
5   "nameIssueType": "Support",  
6   "timespent": null,  
7   "updated": "2023-05-03T09:49:10.742+0200",  
8   "description": "big noob",  
9   "status": "Waiting for customer",  
10  "summary": "noob",  
11  "userId": "qm:2254bb2d-80ac-40c0-b4e8-4ee36123bb1f:7f6f845d-14b0-4b78-8da6-92205f70ea28",  
12  "organizationid": "2",  
13  "createdAt": "2023-05-05T12:48:52.882Z",  
14  "updatedAt": "2023-05-05T12:48:52.882Z"  
15 } ]
```

Back-End

- Worklogs



The screenshot shows a Postman API request configuration. The method is set to GET, and the URL is localhost:8080/api/v1/jira/issue/10014/worklog. The 'Body' tab is selected, showing a JSON response with a pretty-printed structure. The response is an array of worklogs, each containing fields like id, worklog_id, issue_id, creatorId, timespent, updated, description, createdAt, and updatedAt. The first worklog has an id of 5, timespent of 14400, and description "Worklog 4h". The second worklog has an id of 6, timespent of 3600, and description "worklog test". The third worklog has an id of 7, timespent of 7200, and description "test create worklog. Edit : Ajout d'une heure". The fourth worklog has an id of 8, timespent of 3600, and description "test comment worklog". All worklogs were created at 2023-05-04T15:40:22.288Z and updated at 2023-05-04T15:40:22.289Z.

```
[{"id": 5, "worklog_id": "10007", "issue_id": "10014", "creatorId": "70121:a1214ae7-5c7a-422f-8513-edbeb36d6838", "timespent": "14400", "updated": "2023-04-19T15:43:04.689+0200", "description": "Worklog 4h", "createdAt": "2023-05-04T15:40:22.288Z", "updatedAt": "2023-05-04T15:40:22.288Z"}, {"id": 6, "worklog_id": "10009", "issue_id": "10014", "creatorId": "70121:a1214ae7-5c7a-422f-8513-edbeb36d6838", "timespent": "3600", "updated": "2023-04-19T16:07:51.728+0200", "description": "worklog test", "createdAt": "2023-05-04T15:40:22.289Z", "updatedAt": "2023-05-04T15:40:22.289Z"}, {"id": 7, "worklog_id": "10011", "issue_id": "10014", "creatorId": "70121:a1214ae7-5c7a-422f-8513-edbeb36d6838", "timespent": "7200", "updated": "2023-04-19T16:34:05.961+0200", "description": "test create worklog. Edit : Ajout d'une heure", "createdAt": "2023-05-04T15:40:22.290Z", "updatedAt": "2023-05-04T15:40:22.290Z"}, {"id": 8, "worklog_id": "10013", "issue_id": "10014", "creatorId": "70121:a1214ae7-5c7a-422f-8513-edbeb36d6838", "timespent": "3600", "updated": "2023-04-27T11:08:57.124+0200", "description": "test comment worklog", "createdAt": "2023-05-04T15:40:22.293Z", "updatedAt": "2023-05-04T15:40:22.293Z"}]
```

Content

- ETL Diagram
- Jira-Connector
- Back-End
- Front-End

Avec Jira et VisualStudioCode

Front-End

- Design, interface
- TypeScript
- React

Front-End



Why TypeScript ? And not JavaScript

Why React ? And not Angular

TypeScript

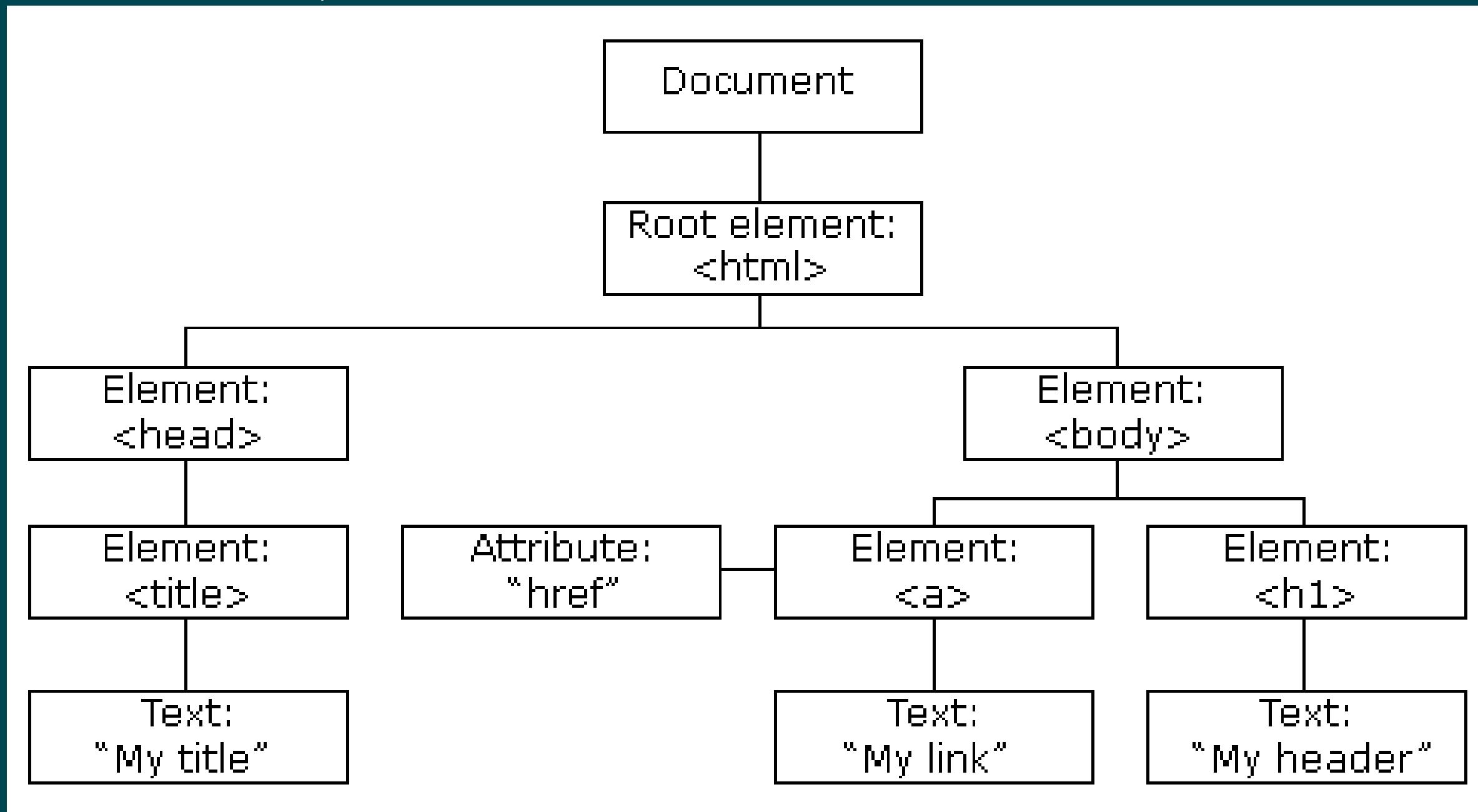
- Programming language build on JavaScript
- Additional syntax to catch errors early
- Static typing : for catching type-related errors at compile time

React

- Building user interfaces
- Usage of components
 - component : Reusable block of code
- .tsx files : To write HTML and TypeScript in the same file

React

DOM (Document Object Model) Tree



React

- Virtual DOM :
 - lightweight copy of the actual DOM is created
 - React updates the virtual DOM, then compares it to the previous version and selectively updates the actual DOM
 - for dynamically update components

React

- **Store (object) :**
 - Holds the whole state tree of your application.
- **Dispatch (method) :**
 - Dispatches an action. This is the only way to trigger a state change.

Front-End

- Before :

The screenshot shows a web application interface for managing users. At the top, there is a dark header bar with the text "Jira Connector" and navigation links for "Users", "Organizations", "Customers", "Add new ▾", "Login", and "Register". Below the header is a search bar labeled "Search by accountId" with a "Search" button. Underneath the search bar is a dropdown menu for "Users per page" set to "5". The main content area is titled "Users List" and contains a message "Please click on a User...". Below this message is a table with five rows, each representing a user. The first row contains the name "Victor Vo". The second row contains "Lucky Luke". The third row contains "Dalton". The fourth row contains "Victor Vo". The fifth row contains "Joe". The first cell in each row is highlighted with a blue background.

Victor Vo
Lucky Luke
Dalton
Victor Vo
Joe

Front-End

- Now :

The screenshot shows the Jira Connector application interface. At the top, there is a dark header bar with the Jira Connector logo, navigation links for Issues, Users, Organizations, Customers, and Add new, and user account links for Login, Register, and Settings.

Below the header is a search bar labeled "Search by organizationID" with a "Search" button. Underneath the search bar is a dropdown menu for selecting the number of organizations per page, currently set to 5.

The main content area is titled "Organizations List". A blue button labeled "1" indicates the current page. Below the page number is a table listing two organizations:

Name	id	Actions
ECAM	1	Edit View customers
BySystems	2	Edit View customers

At the bottom left, there is an "Export organizations list as : .xlsx" dropdown menu and a green "Exporter en xlsx" button.

The footer of the page contains the copyright notice "ECAM © 2023. All rights reserved." and the version information "v0.1.0".

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

React vs Angular

- React : JavaScript Library
- Angular : Front-end Framework
- React : Use of virtual DOM
- Angular : Use of real DOM

Back-End

- Cors

```
var corsOptions = {  
  origin: true,  
  Method: ['POST', 'GET']  
};
```

origin: ["http://localhost:8082", "http://localhost:8081"]

origin: ["http://localhost:8082"]

Search by accountId

Customers per page: 5

Customers List

Please click on a Customer...

1
Youssef Boujraf
Van Olffen Victor
Mourad

Organization ID

Name

Network Error

Search by accountId

Customers per page: 5

Customers List

Please click on

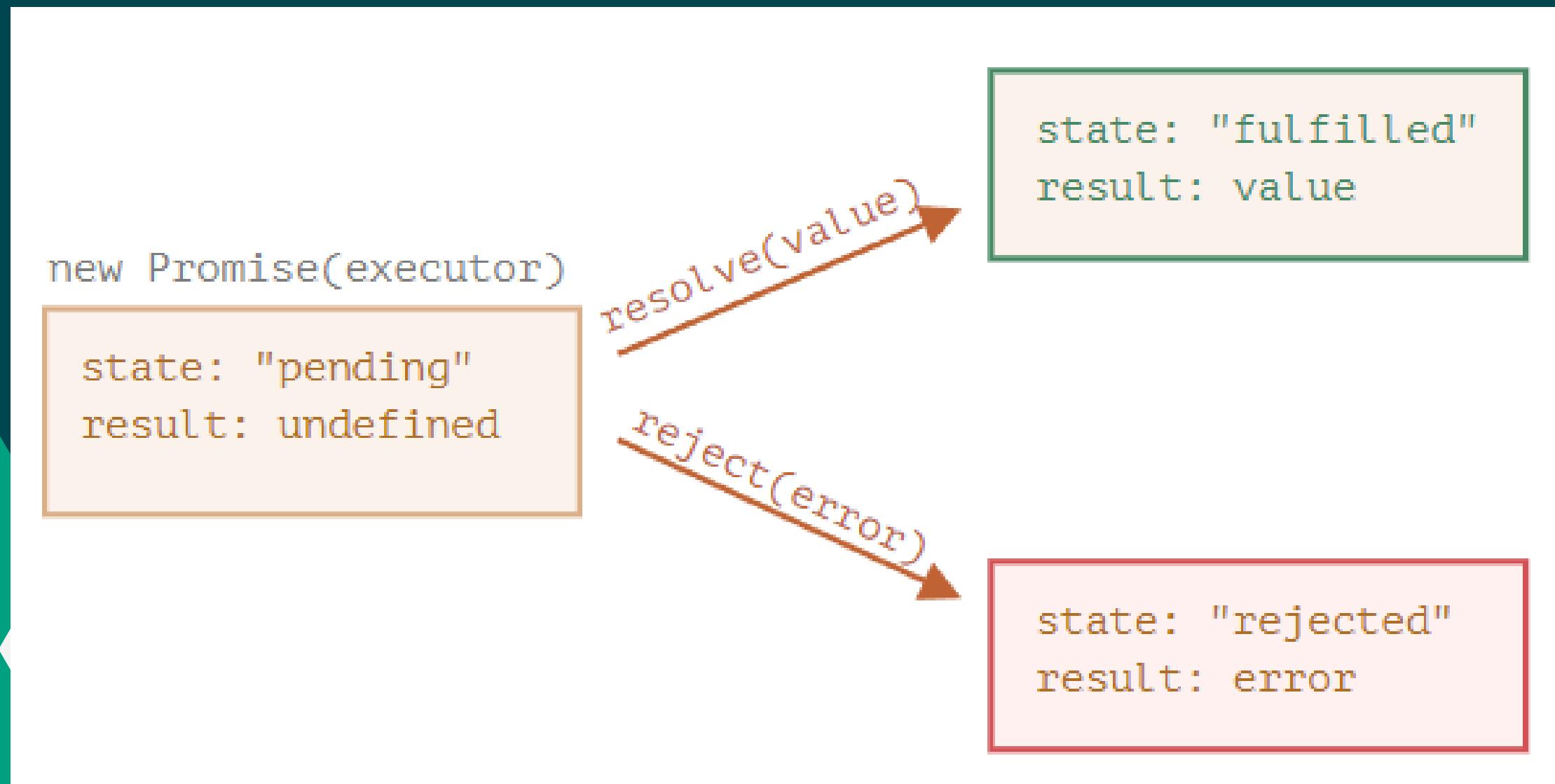
Promise

Utilité

- Permet de continuer d'exécuter le code
- Dire quand elle donnera sa réponse
- Donne la promesse aux parties du code qui en ont besoin

Promise

- 3 états :



Promise

.then

- Promise résolue donne sa réponse

.finally

- Plus nécessaire => cleanup

Axios vs Fetch

- Axios : Données convertis automatiquement en JSON
- Fetch : Utiliser JSON.stringify ({ })
- Axios : réponse directement en JSON
- Fetch : utilise response.json()
- Axios : est compatible avec plus de browser que fetch
- Axios : compatible dans app web et dans node.js alors que fetch que pour web
- Fetch : est une API intégrée => pas d'import ou d'installation

React vs Angular

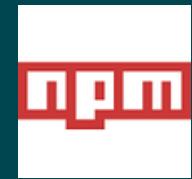
- React est une bibliothèque JavaScript, alors qu'Angular est un framework front-end
- React utilise la liaison de données unidirectionnelle et un virtual DOM (Document Object Model)
- Angular utilise la liaison de données bidirectionnelle et un real DOM

Jira-Connector

Pistes d'amélioration

- Webhooks : Utilisation d'un server avec parefeu au lieu de Ngrok (port forwarding)
- Http requests : Utilisation d'api token pour récup adresse email
- Webhooks : Utilisation asynchrone

Extensions



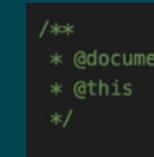
npm Intellisense



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

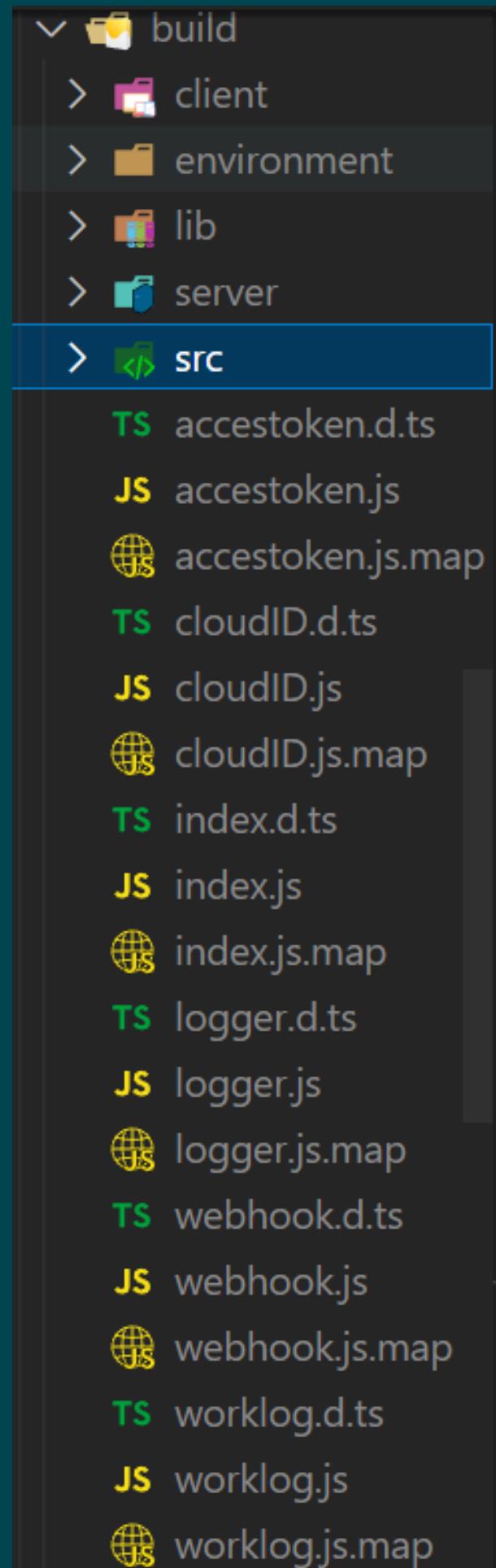
- Crédit à la création de script pour faire du Webhook, des requêtes HTTP, créer des data modèles d'interface.

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,| You, 38  
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

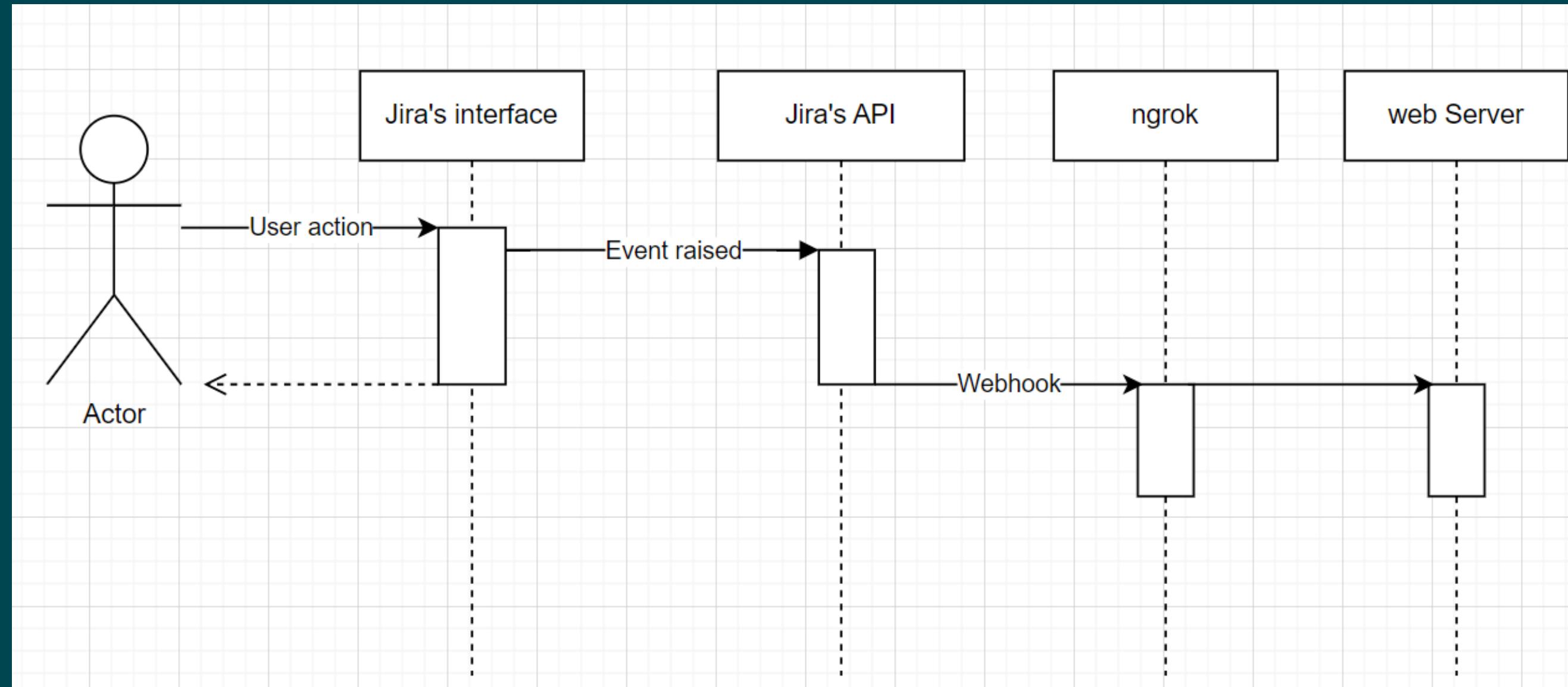
- Extensions
- Packages
- WebHook
- HTTP requests

Avec Jira et Visual Studio Code

Webhook

Web Server : localhost by Express

API : interface de programmation d'application

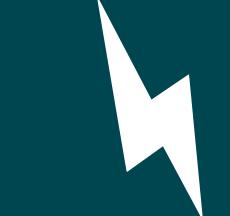


Webhook

lien de rappel HTTP

CRUD

Jira's API



Webhook

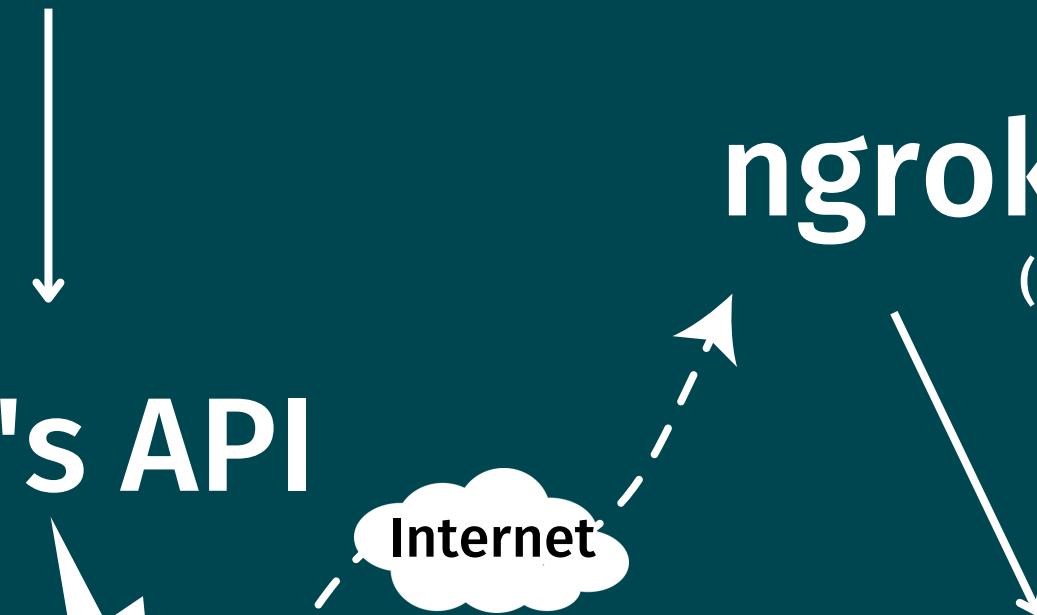
scope : • user
• issues
• comments

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";
```

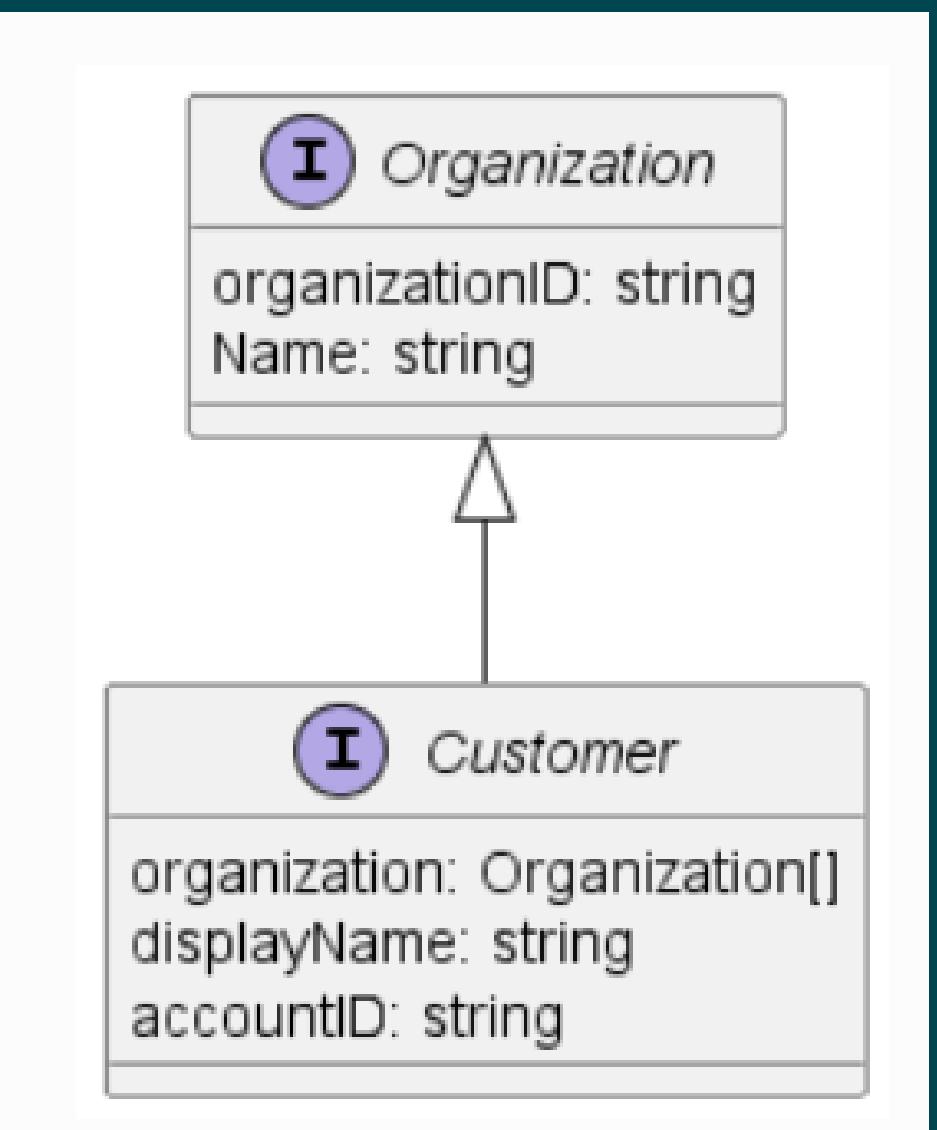
HTTP Requests

- 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] ]  
  }  
]
```

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 buttons: 'Pre-request Script', 'Tests', 'Variables', and 'Runs'. A descriptive text block 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, a dropdown menu is set to 'OAuth 2.0'. A note below explains: 'The authorization data will be automatically generated when you send the request. Learn more about [authorization](#)' followed by a right-pointing arrow. At the bottom, there are two dropdown menus: 'Add auth data to' and 'Request Headers'.

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
- 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

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"  
  }]  
]
```

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}}

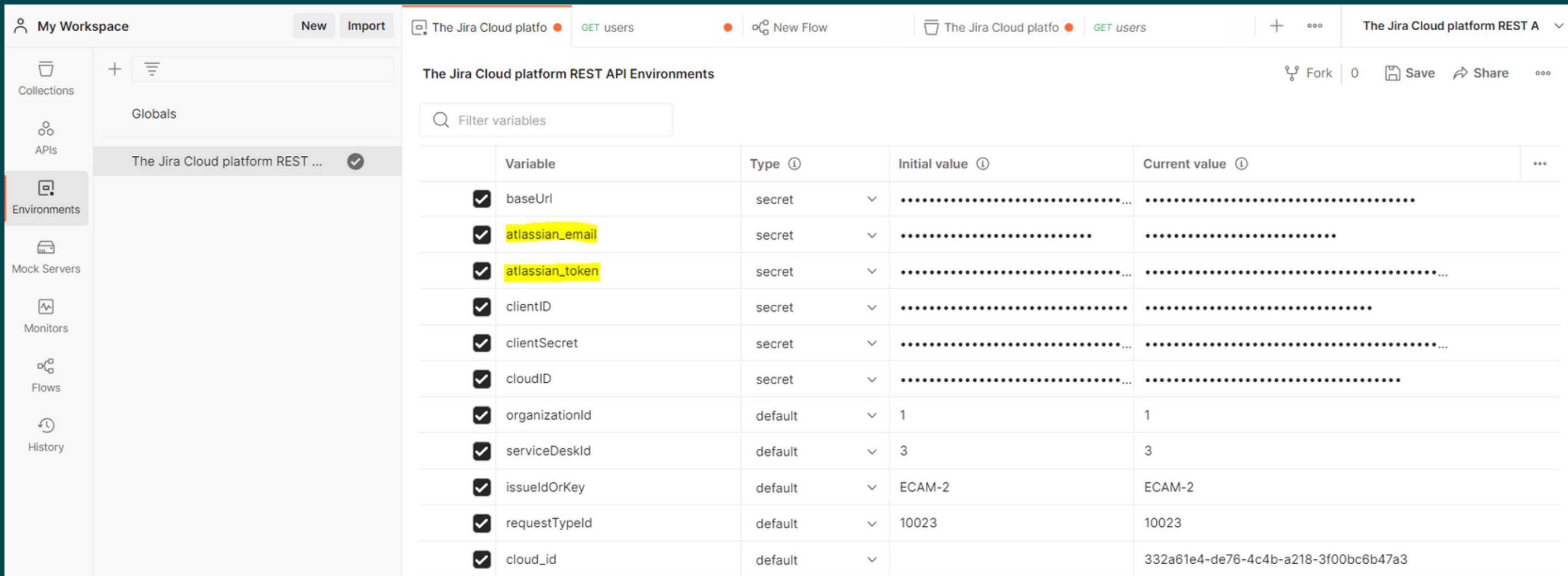
Password {{atlassian_token}}

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

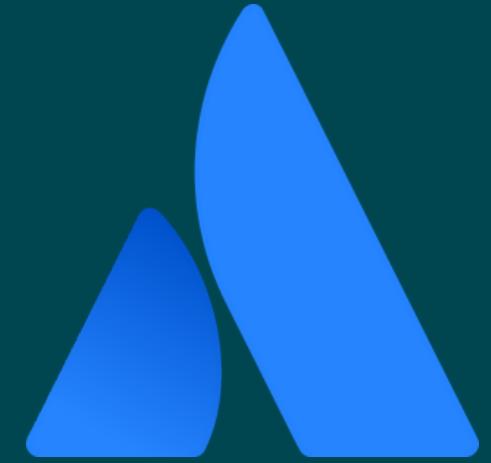
- Version majeur (réécriture complète)
=> 2.0 n'est pas rétrocompatible avec 1.0

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

OAuth 2.0

- Procédure

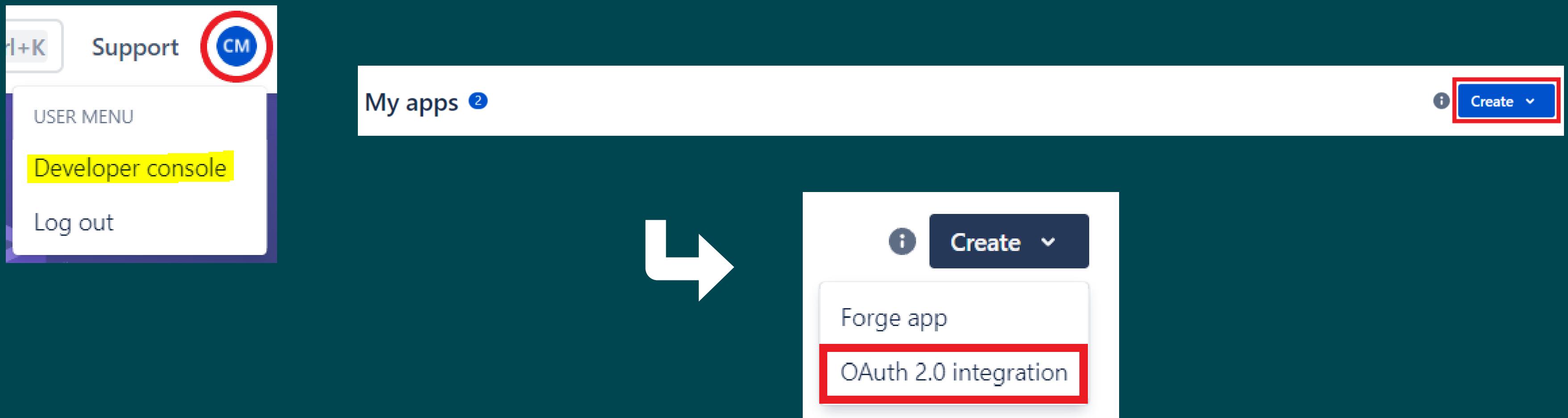
Atlassian Developer



- 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`

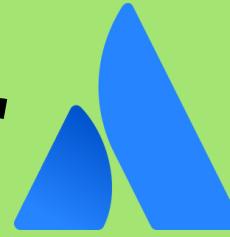
`{{clientId}}`

`{{clientSecret}}`

e.g. read:org

State

Atlassian Developer



- Configuration des permissions

The screenshot shows the 'Postman' application interface. On the left, there is a vertical navigation menu with the following items: POSTMAN, Overview, Distribution, Permissions (which is highlighted with a blue bar), Authorization, and Settings.

The screenshot shows the 'Permissions' section of the Atlassian developer console. The URL in the top navigation bar is 'Console / My apps / Postman / Permissions'. The page title is 'Permissions'.

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

Section: Scopes Used
18

Section: 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	Documentation
User identity API Get the profile details for the currently logged-in user, such as the Atlassian account ID and email.	2	Config...	Documentation
Confluence API Get, create, update, and delete content, spaces, and more.	0	Add	Documentation
BRIE API Create, cancel, and read backup and restore, retrieve and publish cloud details.	0	Add	Documentation
Jira API Get, create, update, and delete issues, projects, fields, and more.	16	Config...	Documentation
Personal data reporting API Report user accounts that an app is storing personal data for.	0	Add	Documentation

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 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 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 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 Read Insight object schemas, objects, and object attributes.	read:servicemanagement-insight-objects

Atlassian Developer

- Permissions pour Jira

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



Choosing your scopes

Use the scopes recommended in the API documentation for the

[Classic scopes](#)

[Granular scopes](#)

Atlassian Developer



- Permissions granular scopes

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

Atlassian Developer



• Activer les permissions

The screenshot shows the 'Authorization' section of the Atlassian API documentation. It includes a sidebar with links: POSTMAN, Overview, Distribution, Permissions, Authorization (which is selected), and Settings. The main content area displays several authorization URLs:

- 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

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
```

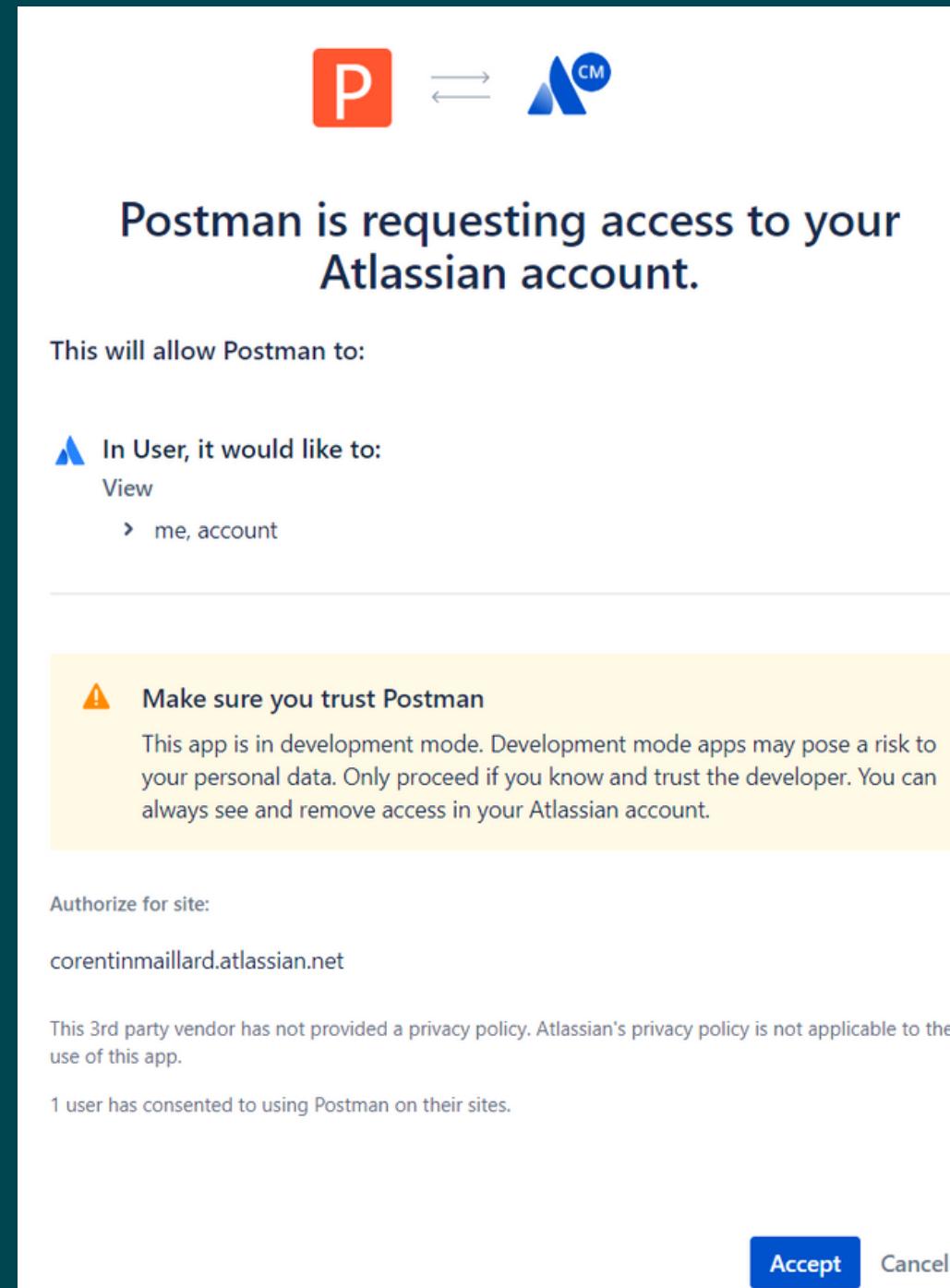


Granular Jira API authorization URL

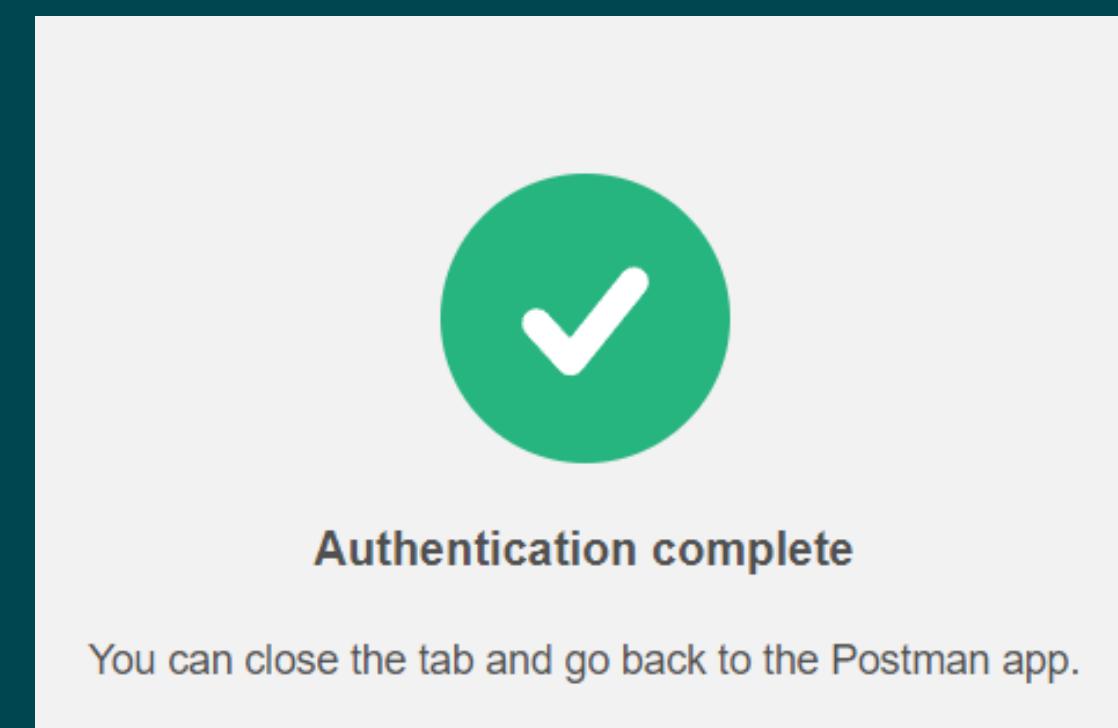
```
https://auth.atlassian.com/authorize?audience=api.atlassian.com&client_id=nr3DZPJ129vipN5AwMPBiAk0AZDBls0I  
&scope=read%3Aapplication-role%3Ajira%20read%3Auser%3Ajira%20read%3Aaudit-log%3Ajira%20read%3Aorgan
```



Atlassian Developer



• Activer les permissions



Atlassian Developer



- Client ID et Client Secret

The diagram illustrates the process of generating OAuth 2.0 credentials. On the left, a white box represents the POSTMAN interface, listing several sections: Overview, Distribution, Permissions, Authorization, and Settings. The 'Authorization' section is highlighted with a blue background. A large white arrow points from this section to the right, where the 'Authentication details' page is shown. This page instructs users to use the Client ID and Secret for authentication, with a link to learn more about OAuth 2.0 integrations. It displays two input fields: 'Client ID' containing the value 'nr3DZPJ129vipN5AwMPBiAk0AZDBIs0I' and 'Secret' containing a redacted value represented by a series of dots. To the right of each input field is a small grey square icon with a double-lined square symbol inside.

POSTMAN

Overview

Distribution

Permissions

Authorization

Settings

→

Authentication details

Use the Client ID and Secret for authentication. [Learn more about OAuth 2.0 integrations.](#)

Client ID

nr3DZPJ129vipN5AwMPBiAk0AZDBIs0I

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 ▼
Access Token URL ⓘ	https://auth.atlassian.com/oauth/token
Client ID ⓘ	nr3DZPJ129vipN5AwMPBiAk0AZDB⚠
Client Secret ⓘ	ATOAA7cwLZ80GoDvHTBInlh-IJgK⚠
Scope ⓘ	e.g. read:org
Client Authentication	Send as Basic Auth header ▼

● Clear cookies ⓘ

Get New Access Token

Postman



- Cloud ID

The screenshot shows the Postman interface. On the left, there's a dropdown menu with "GET" selected. Next to it is a dropdown arrow. To the right of the dropdown is the URL: "https://api.atlassian.com/oauth/token/accessible-resources". To the right of the URL is a blue "Send" button with a dropdown arrow next to it.

```
[  
  {  
    "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. On the left, there's a sidebar with 'My Workspace' containing 'Collections' (JIRA auth 2.0), 'APIs' (The Jira Cloud platform REST API, The Jira Cloud platform REST API Copy), 'Environments', 'Mock Servers', 'Monitors', 'Flows', and 'History'. The main area shows a 'New Import' dialog for 'The Jira Cloud platform REST API'. A 'get user' collection is selected. A 'GET' request is configured with the URL [https://api.atlassian.com/ex/jira/\[cloud_id\]/rest/api/3/users/search?startAt=0&maxResults=50](https://api.atlassian.com/ex/jira/[cloud_id]/rest/api/3/users/search?startAt=0&maxResults=50). The 'Authorization' tab is selected, showing 'Type: OAuth 2.0' and 'Token: jira cloud' with the value `eyJraWQiOiJmZTM2ZThkMzJMTA2N2RjYT`. The 'Headers' tab includes a 'Request Headers' section with 'Header Prefix: Bearer'. The 'Body' tab shows a JSON response with two user objects. The 'Cookies' tab is visible on the right. At the bottom, there's a status bar with 'Status: 200 OK Time: 695 ms Size: 26.8 KB' and a 'Save as Example' button.

```
1 {
2   "self": "https://api.atlassian.com/ex/jira/3e349001-1034-4d3a-900d-75b56a39f784/rest/api/3/user?accountId=712020:eB4b5e29-1c39-4d90-a9c-c50899a22274",
3   "accountId": "712020:eB4b5e29-1c39-4d90-a9c-c50899a22274",
4   "accountType": "atlassian",
5   "avatarUrls": {
6     "48x48": "https://secure.gravatar.com/avatar/131bc91ba141d045a20955540f05193c?d=https%3A%2F%2Favatar-management--avatars.us-west-2.prod.public.atl-paas.net%2Finitials%2FMM-0.png",
7     "24x24": "https://secure.gravatar.com/avatar/131bc91ba141d045a20955540f05193c?d=https%3A%2F%2Favatar-management--avatars.us-west-2.prod.public.atl-paas.net%2Finitials%2FMM-0.png",
8     "16x16": "https://secure.gravatar.com/avatar/131bc91ba141d045a20955540f05193c?d=https%3A%2F%2Favatar-management--avatars.us-west-2.prod.public.atl-paas.net%2Finitials%2FMM-0.png",
9     "32x32": "https://secure.gravatar.com/avatar/131bc91ba141d045a20955540f05193c?d=https%3A%2F%2Favatar-management--avatars.us-west-2.prod.public.atl-paas.net%2Finitials%2FMM-0.png"
10   },
11   "displayName": "Mouad Mettoui",
12   "active": true,
13   "locale": "en_US"
14 },
15 {
16   "self": "https://api.atlassian.com/ex/jira/3e349001-1034-4d3a-900d-75b56a39f784/rest/api/3/user?accountId=5b70c0b80fd0ac85d309f5e9",
17   "accountId": "5b70c0b80fd0ac85d309f5e9",
18   "accountType": "app",
19   "avatarUrls": {
20     "48x48": "https://secure.gravatar.com/avatar/70674fb0f0569dbbd3f2c19ae504462?d=https%3A%2F%2Favatar-management--avatars.us-west-2.prod.public.atl-paas.net%2Finitials%2FCN-5.png",
21     "24x24": "https://secure.gravatar.com/avatar/70674fb0f0569dbbd3f2c19ae504462?d=https%3A%2F%2Favatar-management--avatars.us-west-2.prod.public.atl-paas.net%2Finitials%2FCN-5.png",
22     "16x16": "https://secure.gravatar.com/avatar/70674fb0f0569dbbd3f2c19ae504462?d=https%3A%2F%2Favatar-management--avatars.us-west-2.prod.public.atl-paas.net%2Finitials%2FCN-5.png",
23     "32x32": "https://secure.gravatar.com/avatar/70674fb0f0569dbbd3f2c19ae504462?d=https%3A%2F%2Favatar-management--avatars.us-west-2.prod.public.atl-paas.net%2Finitials%2FCN-5.png"
24   },
25   "displayName": "Chat Notifications",
26   "active": true
27 },
28 {
29   "self": "https://api.atlassian.com/ex/jira/3e349001-1034-4d3a-900d-75b56a39f784/rest/api/3/user?accountId=557058:c6c04dfb-be04-46eb-98a4-e567aa5332c6"
30 }
```

Start working with your team. Next: Invite at least 1 person. invite 67%

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 search bar at the top right. Below it, there are tabs for New, Import, webhook, and New Flow. A large red box highlights the 'New Flow' tab, which is also labeled 2. To the right of the tabs, there's a button labeled 'New Flow' with a red box around it and the number 3. The main workspace is currently empty, with a placeholder text 'Click and drag to add new blocks'. At the bottom center, there's a button labeled 'Start' with a gear icon and a text input field with 'Aa'.

Click and drag to add new blocks

Start Aa

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. 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 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 purple banner with the text "See how Jira, Confluence, and Trello can help your team achieve its biggest goals together at Team '23". A search bar and navigation tabs for "Products" and "Plans" are also present. The main area displays a grid of product links:

Switch to	Product	Owner
	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 is a search bar with "Find by name" and dropdowns for "Products" and "Plans". Below the search bar, a message says "Showing 4 results out of 4 products". A table lists four products:

Product	Plan	Users
Confluence https://mourad.atlassian.net/wiki	Free	1
Jira Administration https://mourad.atlassian.net		
Jira Service Management https://mourad.atlassian.net		
Jira Software https://mourad.atlassian.net		

A context menu is open over the "Jira Administration" row, with the option "Open Jira Administration" highlighted with a red oval. Other menu items include "Manage users", "Manage product access", "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".

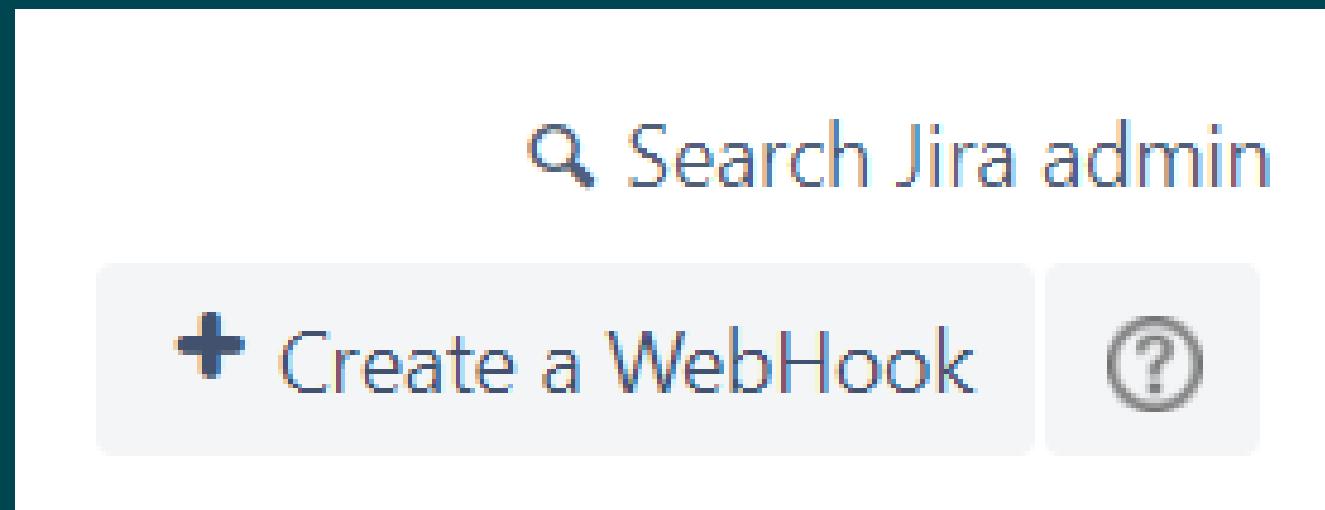


The screenshot shows the Jira Administration screen. At the top, there is a search bar with the text "webhooks" and a close button. Below the search bar, there is a section titled "ADMINISTRATION" with a link "WebHooks" (highlighted with a blue box). At the bottom, there is a note: "Pressing period (.)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

Webhook



- Crée 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 the configuration page for a "New 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.

Name *	New Webhook Listener
Status *	Enabled
URL *	

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

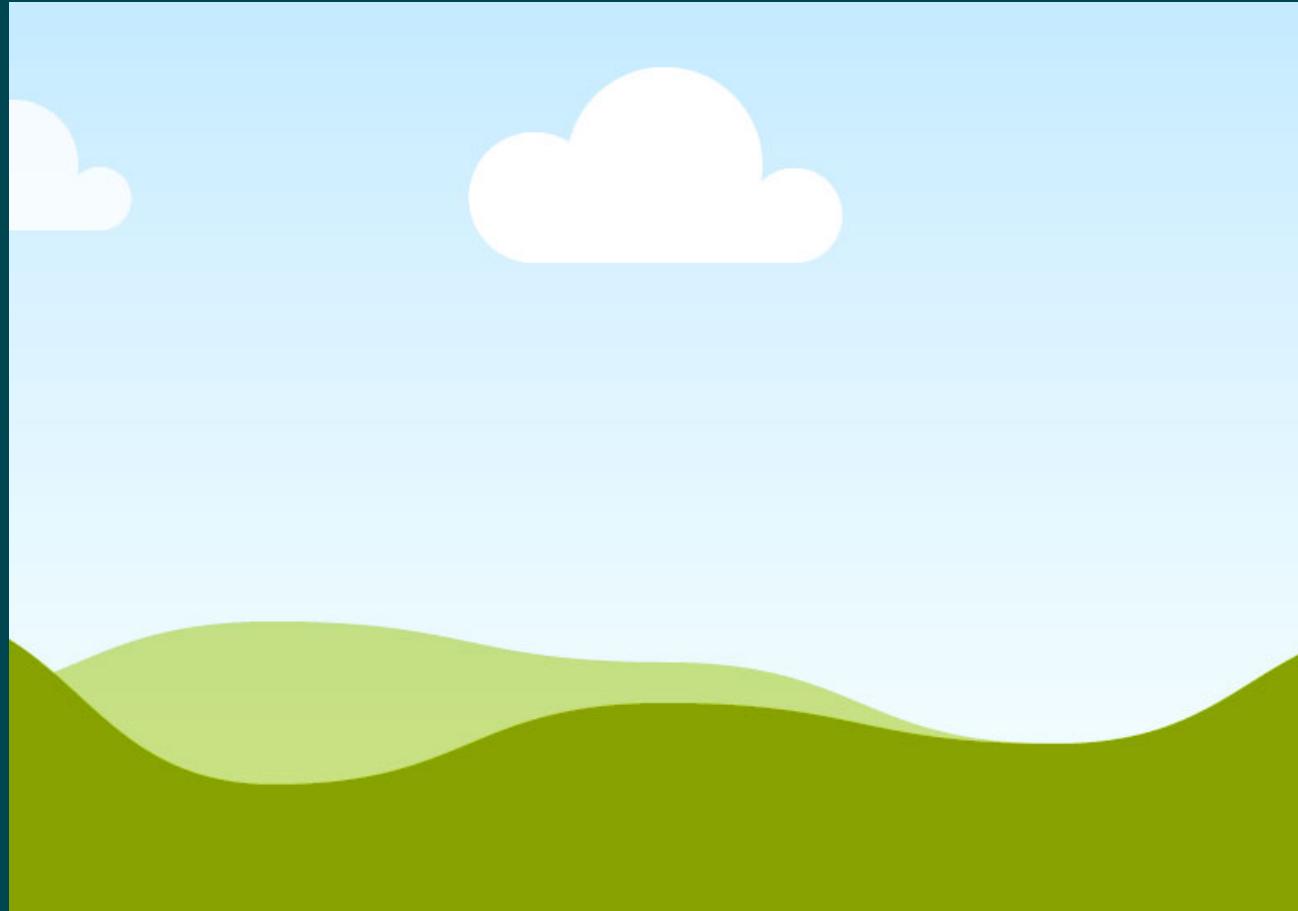
Filter

- created
- updated
- deleted

Webhook



- Configuration du webhook : User

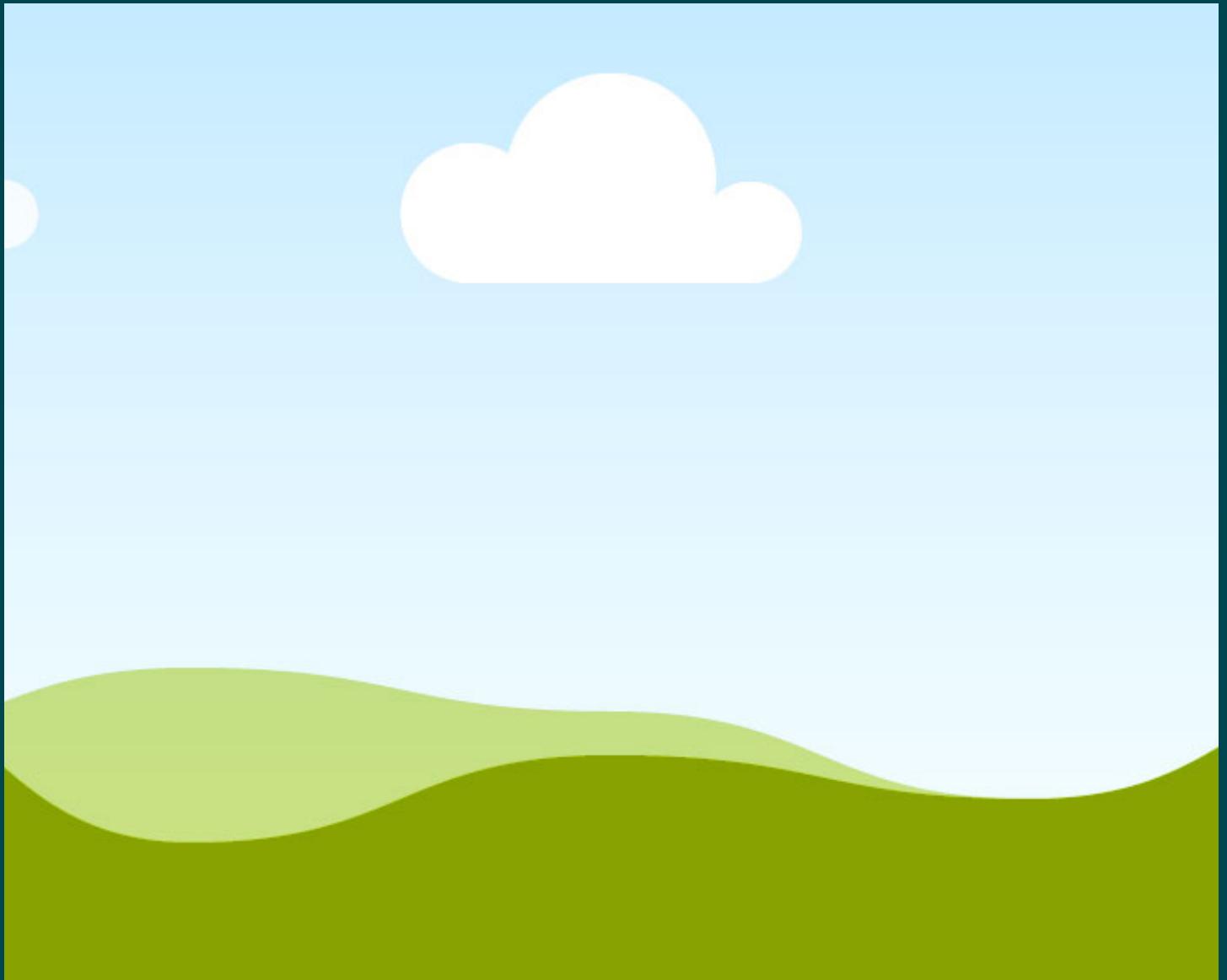


- Pour voir les évènements liés aux utilisateurs

Webhook



- Configuration du webhook : User



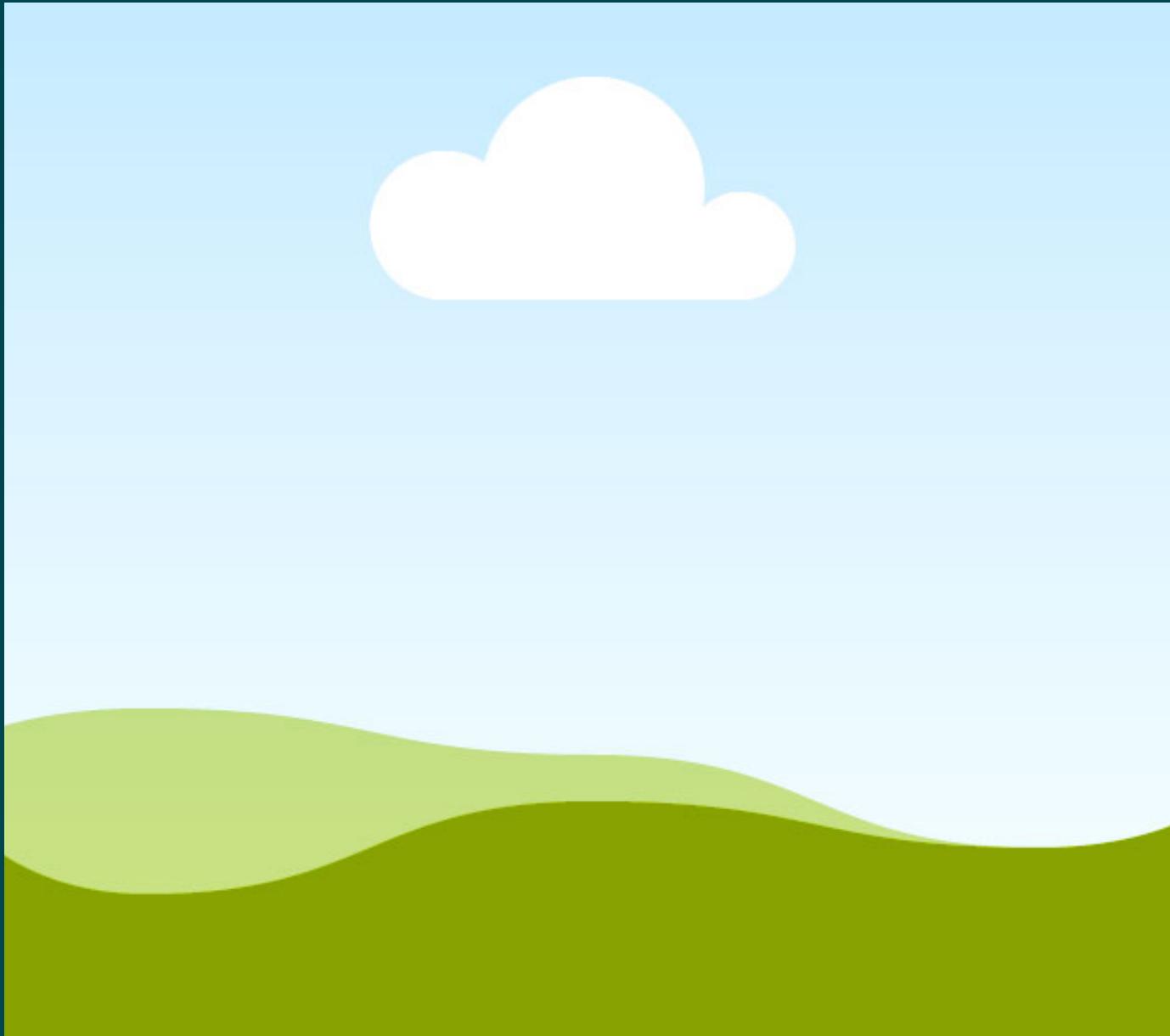
Voir les :

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

Webhook



- Configuration du webhook : Project

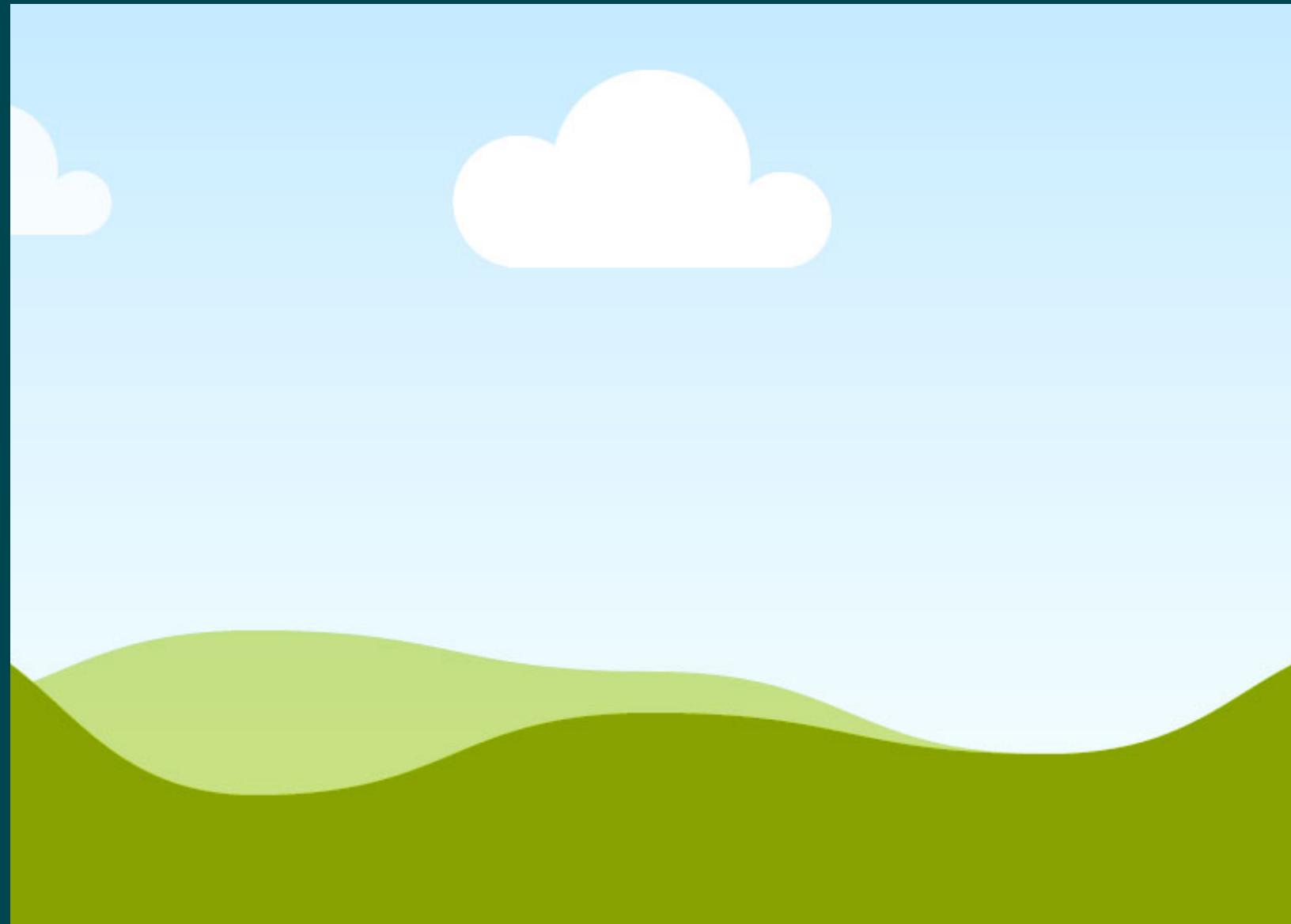


- Rien n'est mit dans les projects : un seul projet

Webhook



- Configuration du webhook : Jira



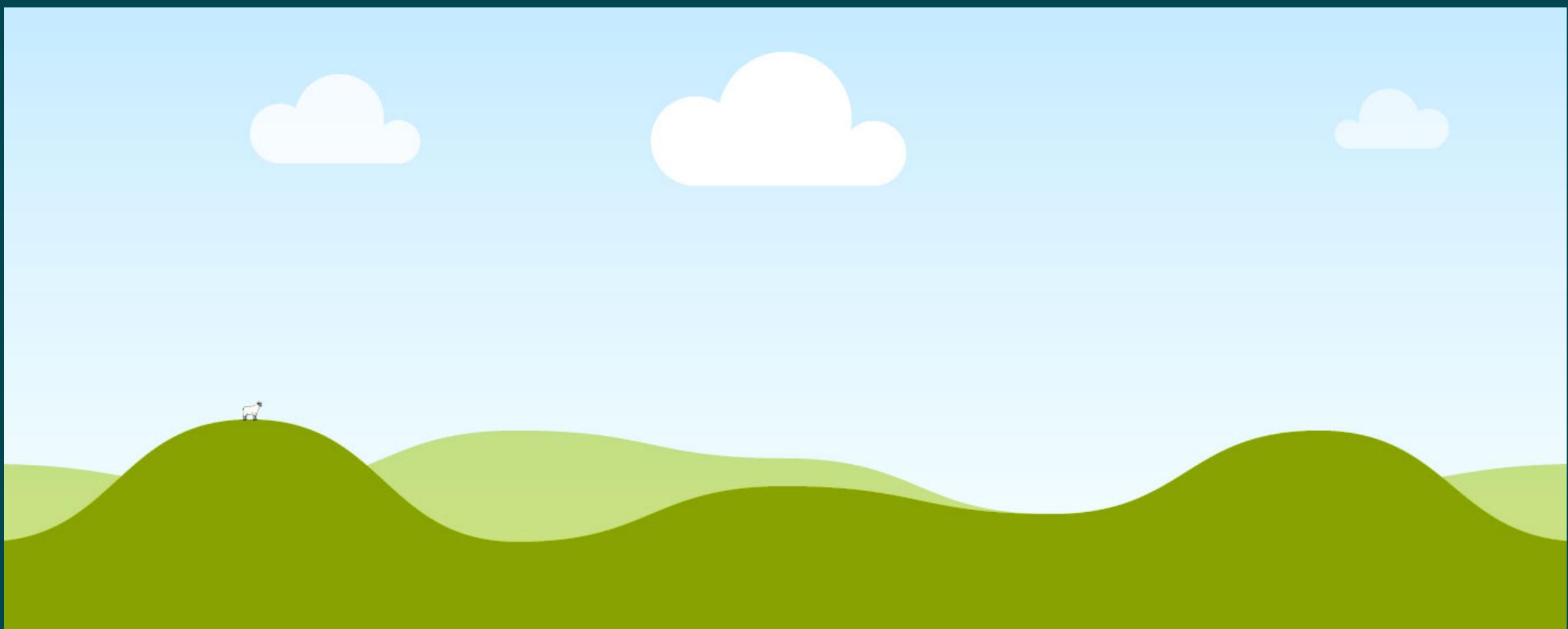
- Pas de sprint
- Pas de board
- Créez le webhook



Webhook



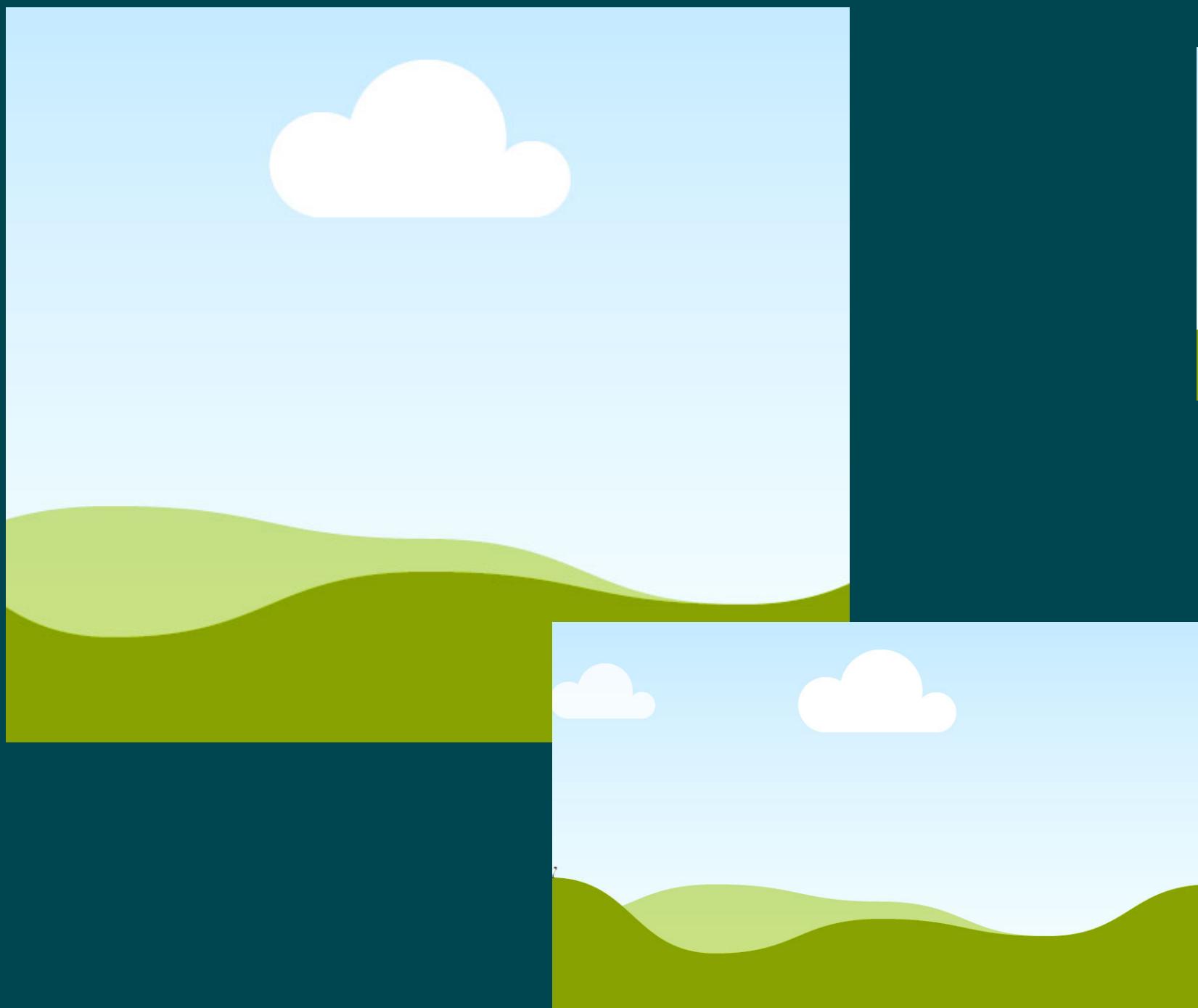
- Configuration du webhook : Jira



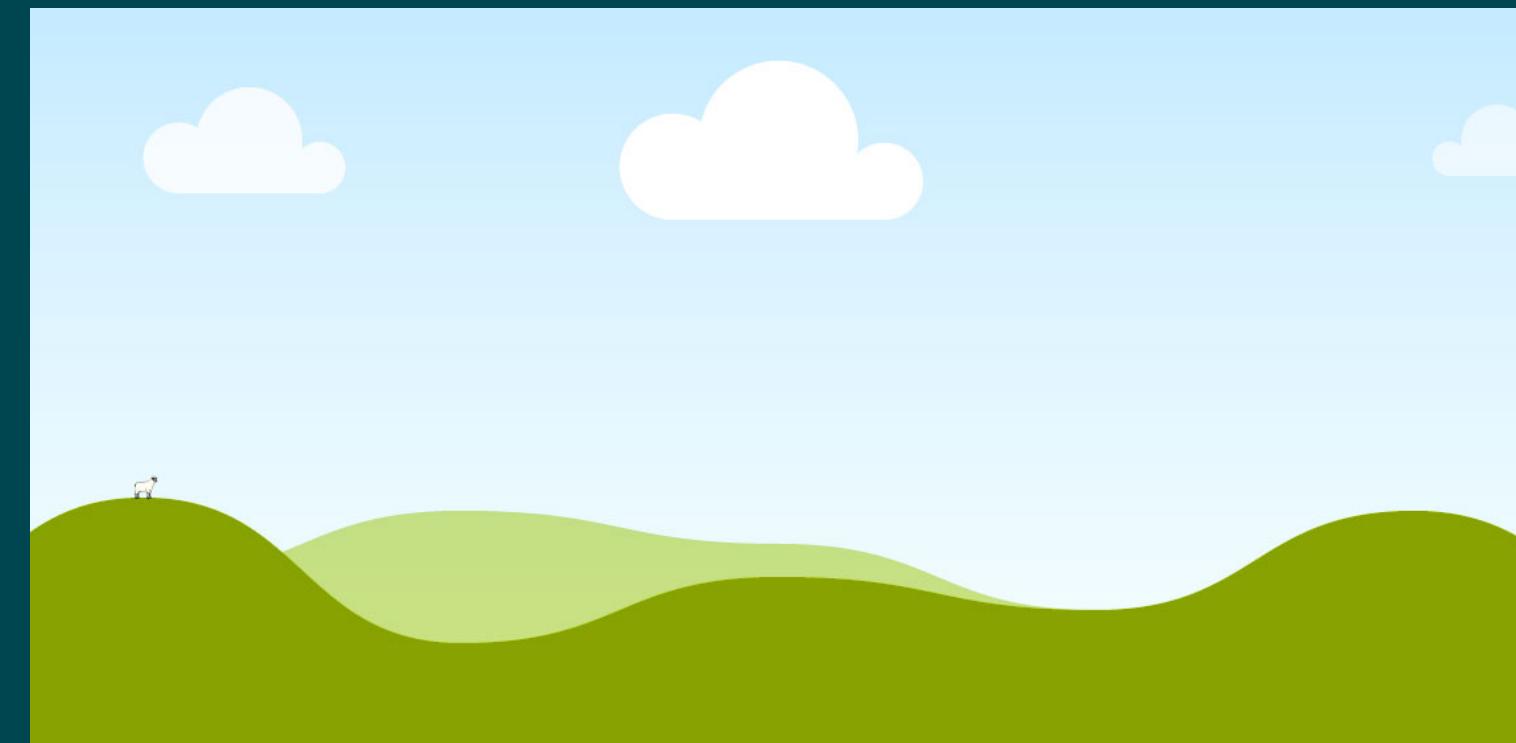
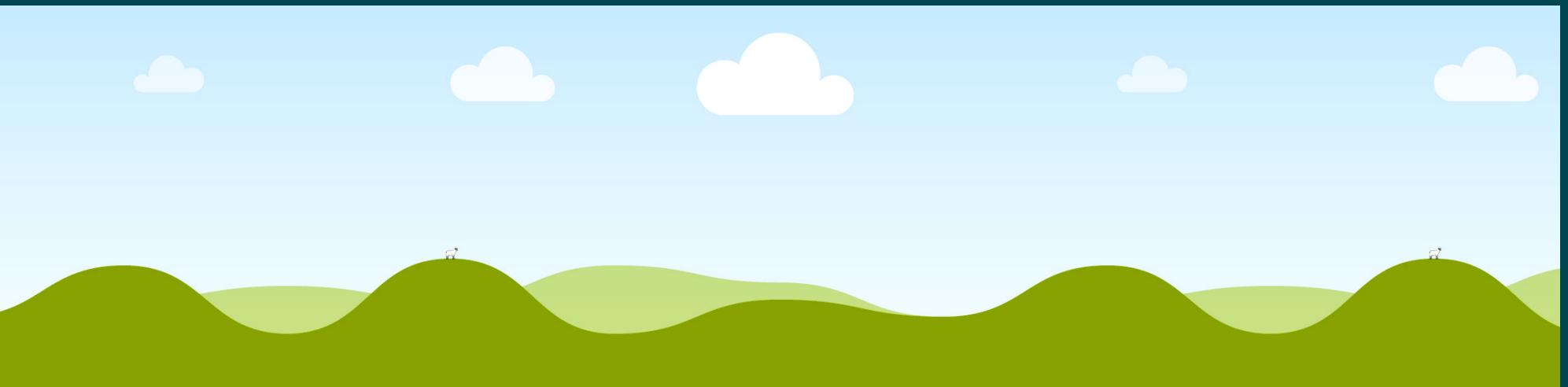
Webhook



- dans issue de Jira



- dans Postman



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

Presentation

WEEK 5

Van Offen Victor
Maillard Corentin
Mettioui Mourad

