Auth0 + Spring Security

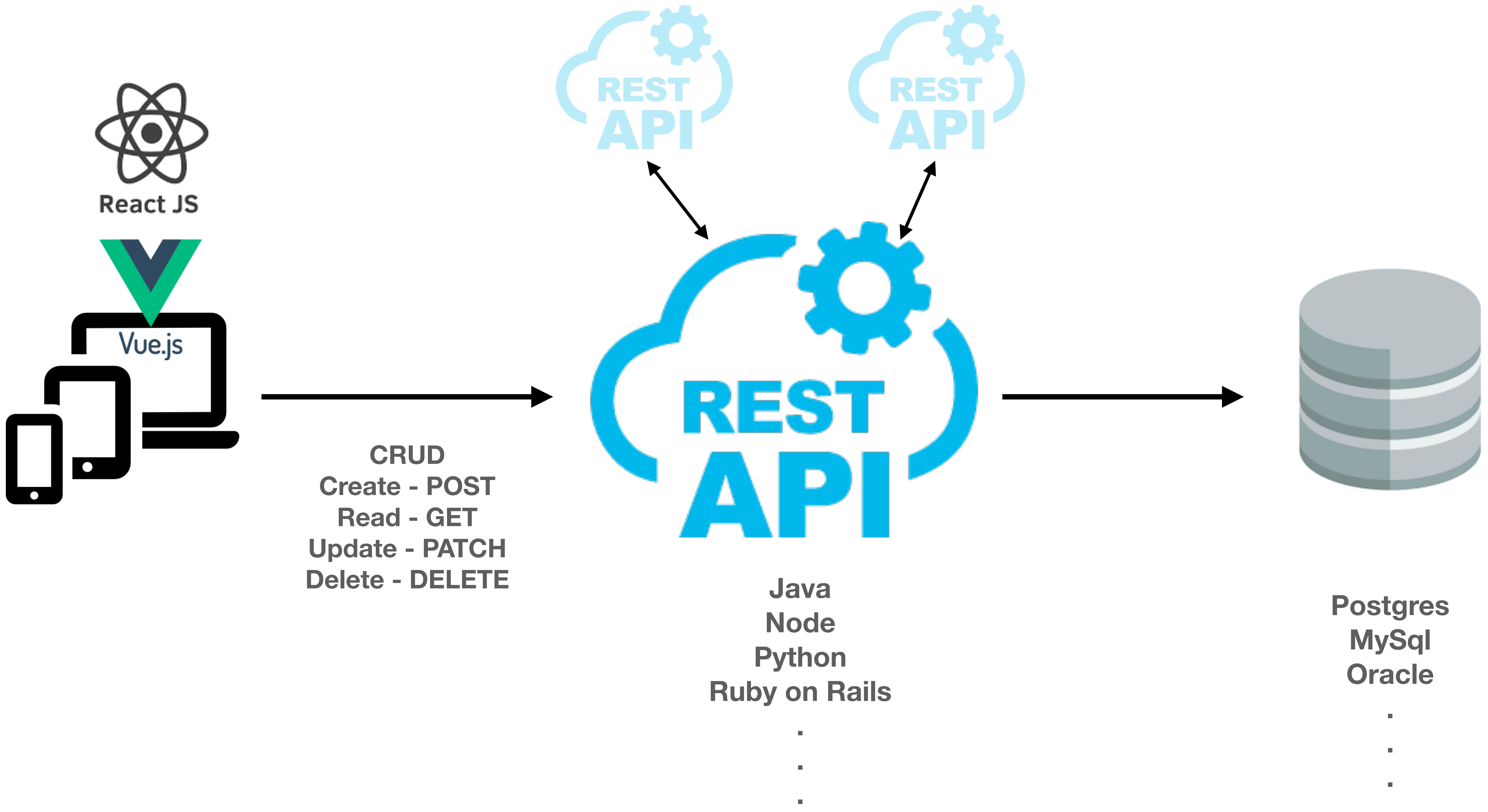
2023.4.18 Kim DoHyung

Contents

- Rest API
- 認証・認可
- Spring Boot Install
- Auth0の設定
- Spring Securityの設定
- Postmanで実行

Rest API

Representational State Transfer
Application Programming Interface



CRUD
Create - POST
Read - GET
Update - PATCH
Delete - DELETE

Java
Node
Python
Ruby on Rails
.
.
.

Postgres
MySql
Oracle
.
.
.

認証・認可

Authentication : 認証 : 入場券

Authorization : 認可 ：乗車券

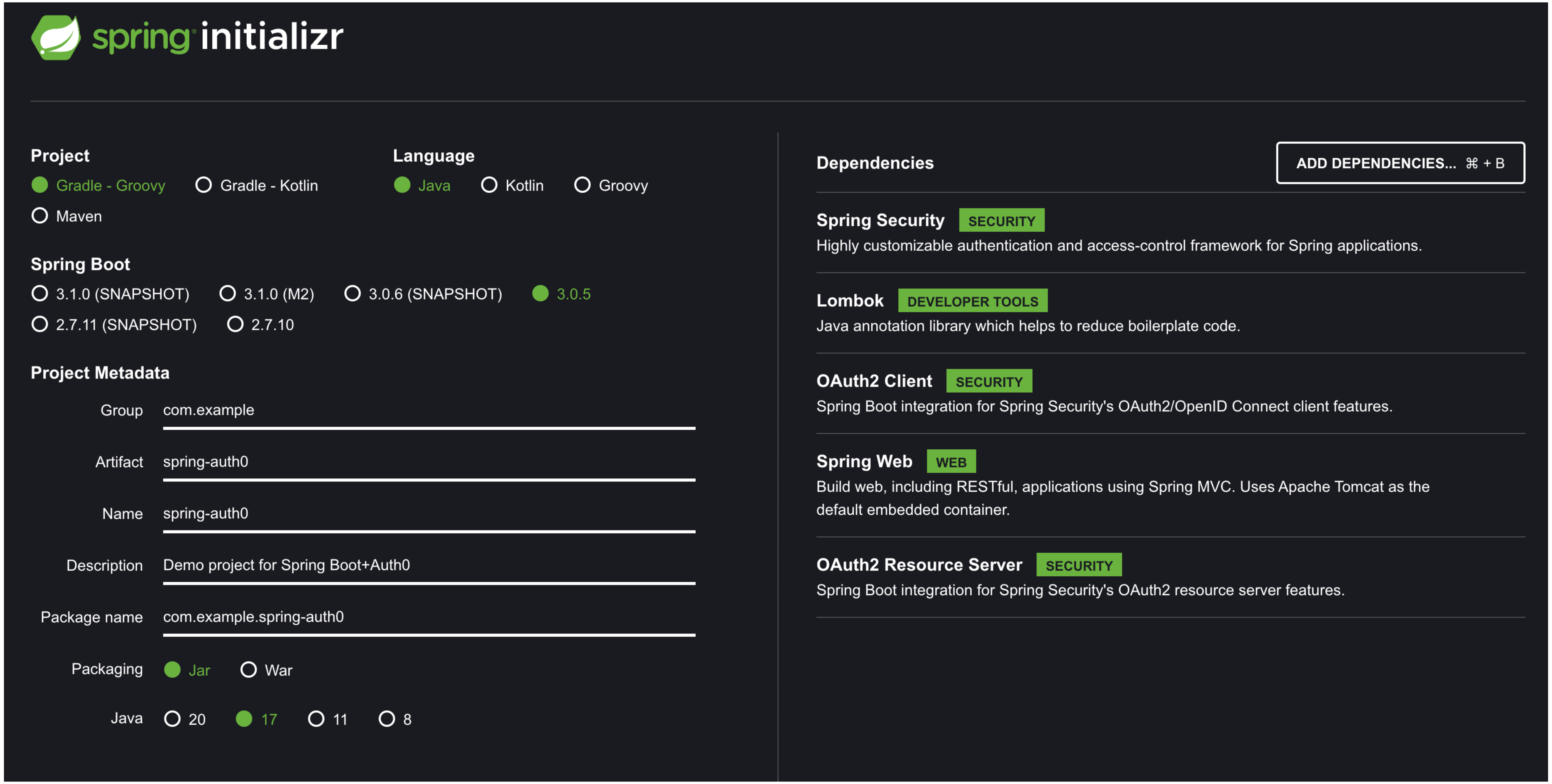
REST API

Authentication
認証
入場券

Authorization
認可
乗車券

Spring Boot

https://start.spring.io/



Project

● Gradle - Groovy ○ Gradle - Kotlin
○ Maven

Language

● Java ○ Kotlin ○ Groovy

Spring Boot

○ 3.1.0 (SNAPSHOT) ○ 3.1.0 (M2) ○ 3.0.6 (SNAPSHOT) ● 3.0.5
○ 2.7.11 (SNAPSHOT) ○ 2.7.10

Project Metadata

Group: com.example

Artifact: spring-auth0

Name: spring-auth0

Description: Demo project for Spring Boot+Auth0

Package name: com.example.spring-auth0

Packaging: ● Jar ○ War

Java: ○ 20 ● 17 ○ 11 ○ 8

Dependencies

Spring Security SECURITY
Highly customizable authentication and access-control framework for Spring applications.

Lombok DEVELOPER TOOLS
Java annotation library which helps to reduce boilerplate code.

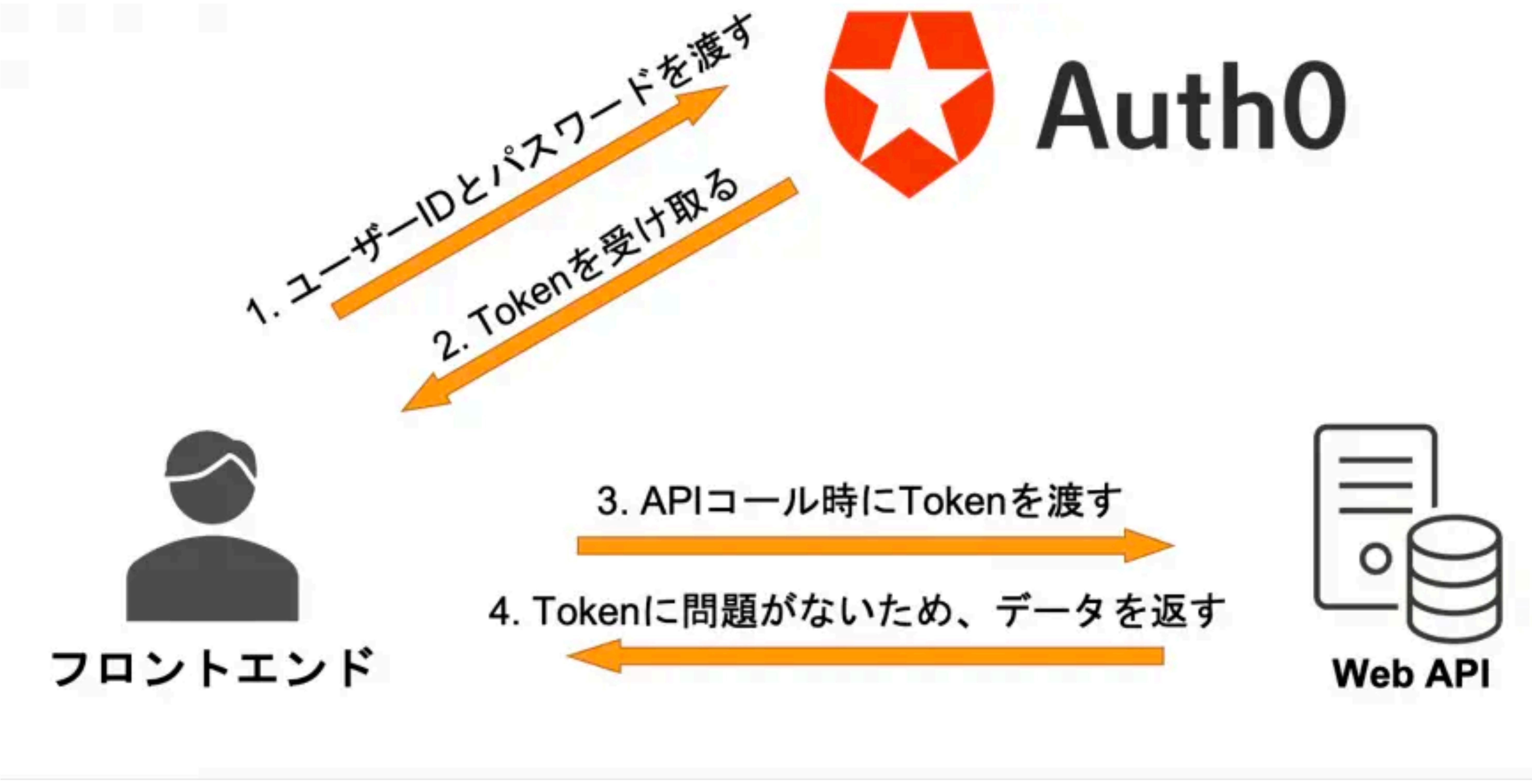
OAuth2 Client SECURITY
Spring Boot integration for Spring Security's OAuth2/OpenID Connect client features.

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

OAuth2 Resource Server SECURITY
Spring Boot integration for Spring Security's OAuth2 resource server features.

Auth0設定

https://auth0.com/



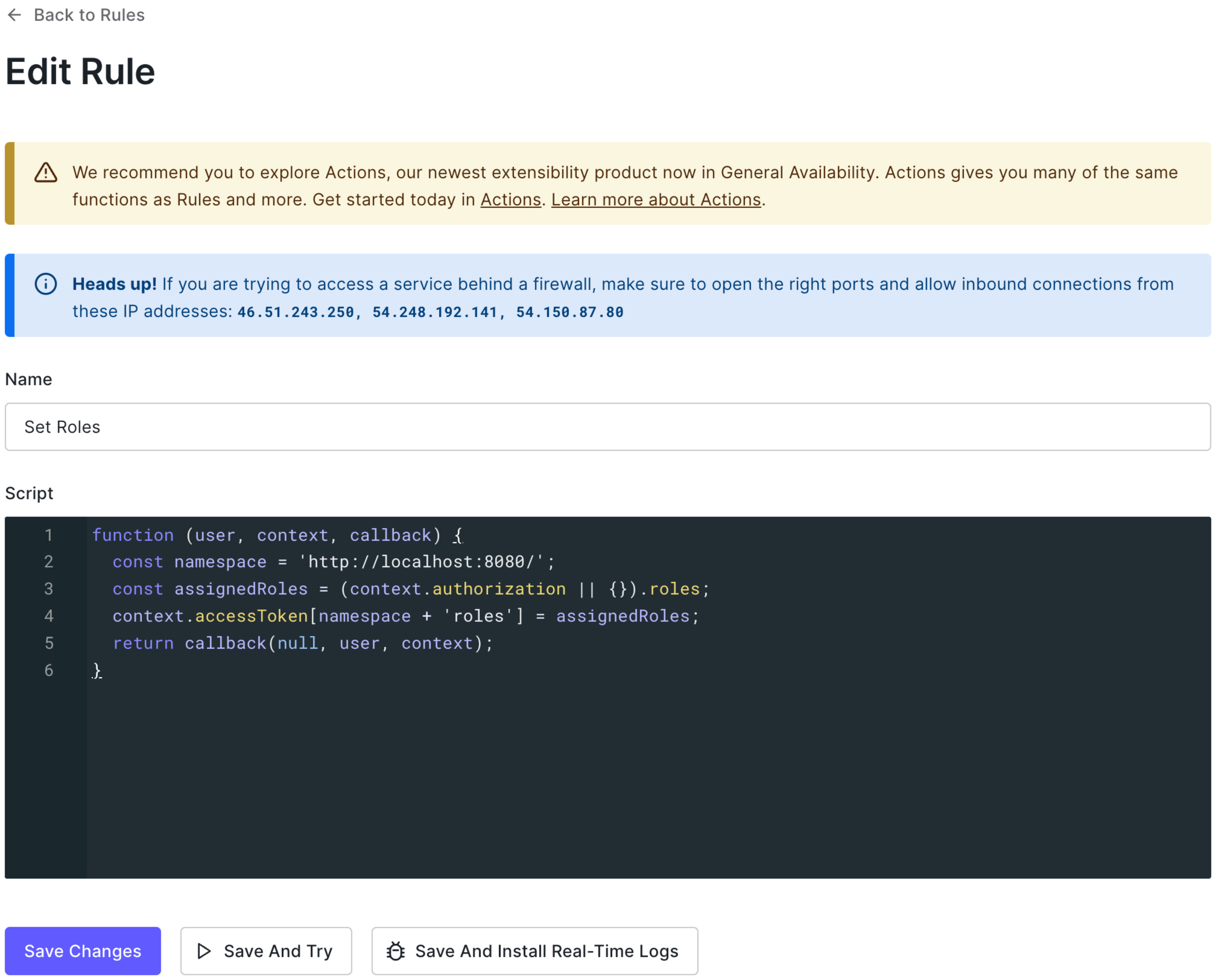
1. ユーザーIDとパスワードを渡す
2. Tokenを受け取る
Auth0
3. APIコール時にTokenを渡す
4. Tokenに問題がないため、データを返す
フロントエンド
Web API

Auth0設定

https://auth0.com/

1. Applications > APIs > Create API
2. Applications > Applications > Create Application(Regular Web Applications)
3. 作成したApplication > Advanced Settings > Grant Types > Passwordをチェック
4. User Management > Roles作成
5. User Management > ユーザ作成&Role付与
6. Setting > General > API Authorization Settings > Default Directoryに「Username-Password-Authentication」を入力する
7. Auth Pipeline > Rulesの作成(Access TokenにユーザのRoleを取り込む処理)

Auth0設定



Edit Rule

We recommend you to explore Actions, our newest extensibility product now in General Availability. Actions gives you many of the same functions as Rules and more. Get started today in Actions. Learn more about Actions.

Heads up! If you are trying to access a service behind a firewall, make sure to open the right ports and allow inbound connections from these IP addresses: `46.51.243.250, 54.248.192.141, 54.150.87.80`

Name: Set Roles

Script

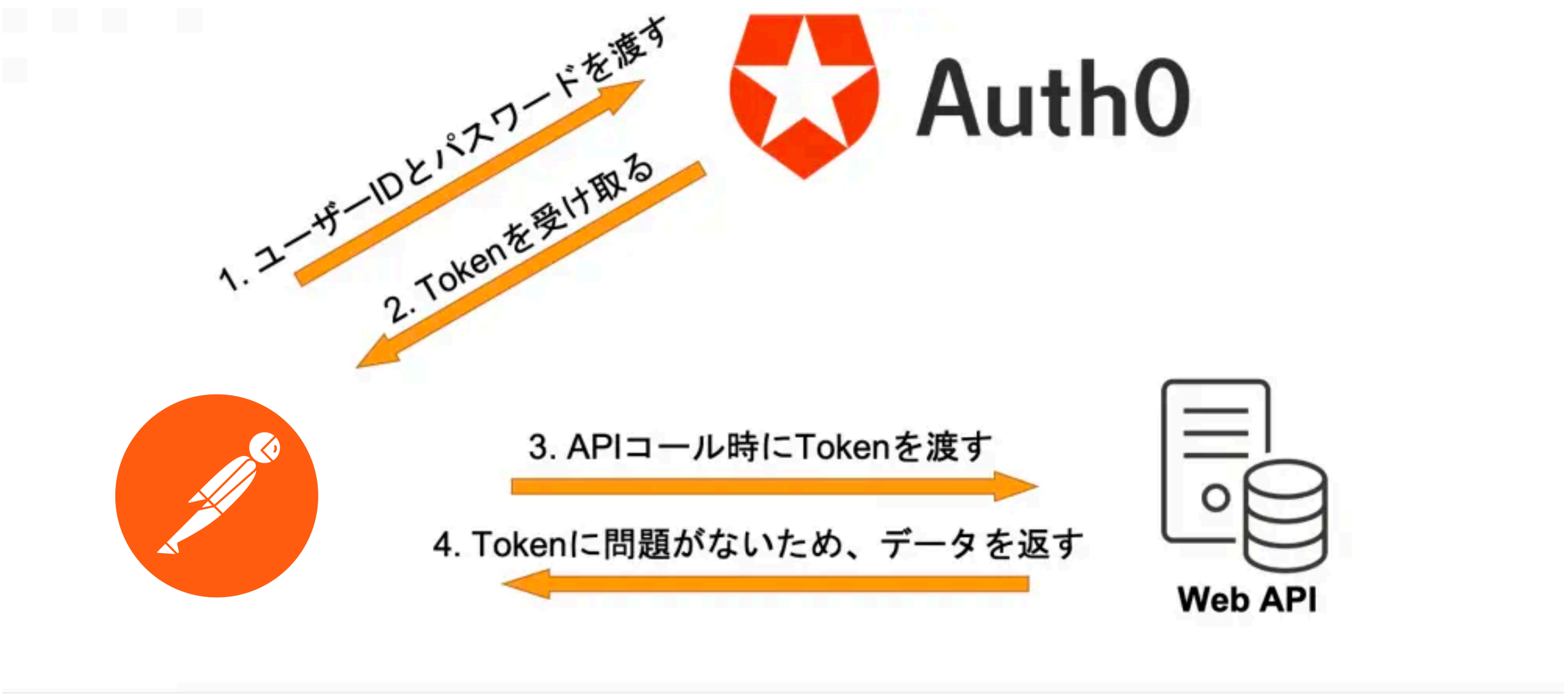
```
function (user, context, callback) {
  const namespace = 'http://localhost:8080/';
  const assignedRoles = (context.authorization || {}).roles;
  context.accessToken[namespace + 'roles'] = assignedRoles;
  return callback(null, user, context);
}
```

```
function (user, context, callback) {
  const namespace = 'http://localhost:8080/';
  const assignedRoles = (context.authorization || {}).roles;
  context.accessToken[namespace + 'roles'] = assignedRoles;
  return callback(null, user, context);
}
```

Postmanで実行

https://www.postman.com/

POSTMAN



1. ユーザーIDとパスワードを渡す

2. Tokenを受け取る

3. APIコール時にTokenを渡す

4. Tokenに問題がないため、データを返す

More..

- フロントエンド(React, VueJsなどとの連携)
- Access TokenとID Tokenの違い
  - https://dev.classmethod.jp/articles/auth0-access-token-id-token-difference/
- ソースコード
  - https://github.com/Kim829/auth0-spring.git

Thank you