**ITÉRATION 6**

Le protocole OAuth 2

| **1.1 — Pourquoi OAuth 2?** |  |
| --- | --- |

|  | **Question** | **Réponse** |
| --- | --- | --- |
| 1 | Peut-il y avoir une raison d’entrer votre mot de passe Google dans une application tierce ? |  |
| 2 | Lequel des éléments ci-dessous représente un risque pour l'utilisateur qui entre directement son mot de passe dans une application ?   1. L'application peut potentiellement accéder et modifier toutes les données de leur compte 2. Le serveur OAuth peut apprendre leur mot de passe. |  |
| 3 | Vrai ou faux : OAuth fournit à l'application l'identité de l'utilisateur qui se connecte. |  |
| 4 | Vrai ou faux : OAuth fournit une méthode pour une application d'obtenir un jeton d'accès afin de réaliser des requêtes API. |  |

**1.2 — Les rôles OAuth 2**

Préciser chaque rôle OAuth2 dans les cas d’usage suivants:

|  | **Cas d’usage** | **Resource Owner (User)** | **User Agent (Device)** | **Client (App)** | **Authorization Server** | **Resource Server (API)** |
| --- | --- | --- | --- | --- | --- | --- |
| 5 | Un développeur utilise Travis CI pour tester son code stocké dans GitHub. |  |  |  |  |  |
| 6 | Vous créez une application iPhone qui utilise le service externe Google pour s’authentifier. |  |  |  |  |  |
| 7 | Vous développez une API en Java qui utilise Okta pour les autorisations. |  |  |  |  |  |
| 8 | Vous connectez Discord au logiciel de surveillance Grafana afin de vous notifier en cas de surcharge dans votre serveur de production. |  |  |  |  |  |

**1.3 — Les types d’applications**

Cocher si c’est des Applications (Clients OAuth) Confidentiels ou Publiques:

|  | **Application** | **Confidentiel** | **Publique** |
| --- | --- | --- | --- |
| 9 | Vous construisez une application Mobile en utilisant Flutter |  |  |
| 10 | Vous construisez une application Mobile en utilisant des technologies Web (http/js/css) |  |  |
| 11 | Une application .NET utilisée par vos clients |  |  |
| 12 | Une application iPhone utilisée par employés |  |  |
| 13 | Une application Desktop native pour gérer l’inventaire |  |  |
| 14 | Une application Java déployé dans Tomcat utilisée pour les publications de billets de blog |  |  |
| 15 | Un outil de ligne de commande CLI qui utilise l’API Github |  |  |
| 16 | Une application Android TV qui utilise l’API Google Maps |  |  |

**1.4 —Enregistrement des Applications (Clients OAuth2)**

Le but de ces deux exercices de cette section est de savoir ce que les équipes de développement de l’application ont besoin d’avoir après l’enregistrement de leurs applications dans le serveur d’autorisation OAuth2 (client\_id, client\_secret), et ce qu’ils doivent renseigner pour que le serveur d’autorisation puisse renvoyer le code d’autorisation une fois l’utilisateur aura donné son consentement (URI de redirection / ou callback).

**Exercice 1:**

Vous devriez créer un nouveau compte dans Auth0.com et enregistrer une nouvelle application de type “Regular Web Applications” (Client confidentiel).   
Notez bien les deux informations d’identification Client\_Id et Client\_Secret, et mettre un URI de redirection dans la configuration de l’application.

Attacher ici une capture d’écran qui montre la bonne création de votre application dans Auth0.com

**Exercice 2:**

Notez ici la preuve de création de l’application dans Dropbox.

Vous pouvez tester le bon fonctionnement en appelant le serveur d’autorisation de dropbox comme suit:

https://www.dropbox.com/oauth2/authorize?client\_id=MY\_CLIENT\_ID&redirect\_uri=MY\_REDIRECT\_URI&response\_type=code

Vous devriez avoir une redirection après consentement de l’utilisateur vers votre URI renseigné avec un paramètre dans l’URL ?code=XXXX (c’est le code d’autorisation).

**1.5 —Authorizations**

Notes sur l’exercice 1 :

|  |
| --- |

Preuve d’accomplissement de l’exercice 2 :

|  |
| --- |

**1.6 — Lab 1: Lister des fichiers dans Google Drive**

Le JSON final de résultat après exécution du flux d’autorisation OAuth:

|  |
| --- |

**1.7 — Lab 2: Intégration Travis CI et GitHub**

Mettre ici une capture d’écran qui montre la bonne intégration (un répo GitHub contenant un YAML Travis-CI et découvert dans celui-ci par exemple :

|  |
| --- |

**1.8 — Lab 3: Discord et ses connections**

Mettre ici une capture d’écran qui montre la bonne intégration et les différents paramètres OAuth2 passer dans l’URL pour activer l’autorisation:

|  |
| --- |

**1.9 —Lab 4: Implémenter soi-même une application cliente OAuth2 (optionnel)**

Mettre ici une description sur la manière dont votre application fonctionne pour s’intégrer avec l’API externe en utilisant OAuth 2.

|  |
| --- |

**1.10 —Lab 5: S’authentifier dans Grafana en utilisant notre compte GitLab**

Vous devriez avoir une instance de Grafana fonctionnelle sur votre machine local ou sur un serveur distant, qui utilise GitLab comme méthode d’authentification.

|  |
| --- |