# 

|  |  |  |
| --- | --- | --- |
| C:\Users\Bamenkombo\AppData\Local\Microsoft\Windows\INetCache\Content.Word\yamo_group_white.png | SPECIFICATIONS |  |
| **[KIDOLE]**  **Spécification fonctionnelle générale** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identification** | | | |
| **Date CREATION** | 03/ 08 / 2020 | | |
| **Date VALIDATION** |  | | |
| **Code projet** | KIDOLE | | |
| **Rédacteur(s)** | Cyrille MOFFO | | |
| **VALIDEUR(s)** |  | | |
| **liste de diffusion** |  | | |
| **Version** | Date | Contenu de la version | Rédacteur |
| 1.0 | 03/08/2020 |  | Cyrille MOFFO |
|  |  |  |  |
|  |  |  |  |

[1 DESCRIPTION DES INFORMATIONS 3](#_Toc47604655)

[1.1 Les principaux concepts métiers 3](#_Toc47604656)

[1.2 Diagramme de classe des principaux concepts métiers 3](#_Toc47604657)

[2 OBJECTIFS METIERS DE L’APPLICATION 3](#_Toc47604658)

[2.1 Les utilisateurs de l’application 3](#_Toc47604659)

[2.2 Les services rendus par l’application 3](#_Toc47604660)

[2.3 Contraintes de service 3](#_Toc47604661)

[2.4 Volumétrie 3](#_Toc47604662)

[3 LISTE DES FONCTIONNALITES PROPOSEES PAR L’APPLICATION 4](#_Toc47604663)

[3.1 Fonctionnalités métiers 4](#_Toc47604664)

[**3.1.1** **Gestion d’un profil utilisateur / athlète.** 4](#_Toc47604665)

[**3.1.2** **Gestion d’une compétition.** 4](#_Toc47604666)

[3.2 Fonctionnalités support 4](#_Toc47604667)

[3.3 Règles de contrôle et de traitement des erreurs 4](#_Toc47604668)

[4 DESCRIPTION DE L’ARCHITECTURE 5](#_Toc47604669)

[4.1 Architecture fonctionnelle et applicative 5](#_Toc47604670)

[**4.1.1** **BACKEND** 5](#_Toc47604671)

[4.2 Exigences sécurité 7](#_Toc47604672)

[5 DEROULEMENT DU TRAITEMENT 7](#_Toc47604673)

[5.1 Données en entrée du service applicatif 7](#_Toc47604674)

[5.2 Règles de transformation des données 8](#_Toc47604675)

[5.3 Règles de contrôles de la donnée 8](#_Toc47604676)

[5.4 Messages d’erreur 9](#_Toc47604677)

[5.5 Données en sortie du service applicatif 10](#_Toc47604678)

*Ce modèle de document permet de spécifier de façon générale les objectifs de l’application, son périmètre, les principes de fonctionnement de cette application en terme de service rendu.*

# DESCRIPTION DES INFORMATIONS

## Les principaux concepts métiers

L’ensemble des principaux concepts qui forment la colonne vertébrale du MCD

* La gestion des profils utilisateurs/athletes
* La gestion des compétitions

## Diagramme de classe des principaux concepts métiers

*Insérer le MCD représentant les relations entre les principaux concepts métiers qui vont former la colonne vertébrale.*

# OBJECTIFS METIERS DE L’APPLICATION

## Les utilisateurs de l’application

Cette application est destinée aux :

* Professionnels sportifs
* …

## Les services rendus par l’application

Les services rendus par l’application aux utilisateurs :

* Maitriser les données sur les compétitions des athlètes
* Faire des statistiques sportives
* Créer une base de données sportives / archives sportives

## Contraintes de service

L’application doit être disponible 24h/24, 7j/7

## Volumétrie

*Décrire le nombre d’utilisateurs susceptibles de se connecter…*

# LISTE DES FONCTIONNALITES PROPOSEES PAR L’APPLICATION

## Fonctionnalités métiers

L’ensemble des fonctionnalités proposées par l’application en terme de service rendu à l’utilisateur :

### **Gestion d’un profil utilisateur / athlète.**

* Inscription/enregistrement d’un compte athlète
* Afficher les informations d’un profil
* Authentification/connexion d’un compte
* Déconnexion d’un compte athlète
* Compléter, modifier, actualiser le profil d’un compte
* Supprimer un compte
* Désactiver un compte
* Modification du mot de passe

### **Gestion d’une compétition.**

* Créer une compétition
* Afficher les détails d’une compétition
* Ajouter des membres à la compétition
* Actualiser les informations de la compétition
* Supprimer une compétition
* Retirer un compte utilisateur/membre de la compétition

## Fonctionnalités support

*Lister l’ensemble des fonctionnalités proposées par l’application en terme*

## Règles de contrôle et de traitement des erreurs

*Décrire les principes de contrôle qui seront mis en place et les conditions de reprise.*

Les logs générés par l’application auront des niveaux différents.

Ci-dessous les différents types de niveaux des logs

* **TRACE** : affichage d’un log pour tracer l’exécution d’une fonction
* **DEBUG** : affichage d’un log de debug
* **INFO** : l’affichage d’un log informatif
* **WARN** : lorsque l’application rencontre une situation exceptionnelle, mais non bloquante pour la suite du processus de traitement
* **ERROR** : lorsque l’application rencontre une situation ou elle est incapable de continue le traitement commencé. Il s’est suit l’arrêt du processus de traitement et retour d’un message d’erreur.

# DESCRIPTION DE L’ARCHITECTURE

## Architecture fonctionnelle et applicative

*Préciser notamment les différents domaines en interface (applications en interface, condition de déclenchement, objet du flux, ...)*

*Préciser les éventuels impacts de la solution retenue sur l'organisation fonctionnelle des applications existantes (nouvelles fonctionnalités, nouvelles informations échangées, modification ou nouveaux flux etc.)*

Ci-dessus, l’architecture fonctionnelle globale de l’application

DB

Flux de données synchrone

Front Office

Backend

Back Office

L’application comporte

* **DB** : la base de données de l’application, qui contiendra toutes les données de l’application
* **Backend** : qui contiendra tout le fonctionnement métier de l’application
* **FrontOffice** : interface web d’accès à l’application destinée au public
* **BackOffice** : interface d’administration de l’application destinée aux administrateurs de l’application

### **BACKEND**

Le backend se compose de plusieurs modules qui sont

* **ms-kidole-accounts :** authentification, gestion des comptes et des profils
* **ms-kidole-competitions :** gestion des competitions

Ces différents modules sont soutenus par les librairies suivantes :

* **lib-kidole-dto :** Librairie ou sont l’ensemble des differents DTO
* **lib-kidole-models :** Librairie ou sont l’ensemble des models (entités)
* **lib-kidole-mapper :** Librairie permettant le mapping entre les entités et les dto
* **lib-kidole-common :** Librairie qui regroupe tous les utilitaires (logging, etc …)

Chaque module a l’architecture logicielle suivante

Model

Consumer

Business

Batch

API

DB

Données entrants

Chaque module est composé

* **Batch** : qui exécute toutes les traitements répétitifs
* **API** : pour gérer les requetés http avec le backoffice et le FrontOffice
* **Business** : pour gérer les traitements métiers
* **Consumer** : pour gérer tous les accès à la base de données

## Exigences sécurité

*Préciser la disponibilité des données et services, la fréquence de sauvegardes des données*

L’infrastructure étant géré par azure (Microsoft), toutes la sécurité des serveurs sur lesquels seront déployés les applications auront ainsi les niveaux de sécurité implémenté par Azure.

L’application backend développée en Java avec le Framework Spring utilisera Spring Security pour la gestion de la sécurité de la partie applicative.

# DEROULEMENT DU TRAITEMENT

## Données en entrée du service applicatif

*Décrire les données en entrée du service applicatif*

En entête de chaque requête, un header est nécessaire.

La structure du header est explicitée ci-dessous

|  |  |  |
| --- | --- | --- |
| *Nom* | *Type* | *Description* |
| X-Ms-Token | String | Clé d’identification attribuée par à chaque session |
| X-Ms-Id | String | Identifiant du compte qui appelle le service |

Les données entrées des services applicatifs sont des objets DTO auxquels sont préfixés les objets ci-dessous

|  |  |  |  |
| --- | --- | --- | --- |
| *Nom* | *Type* | *Obligatoire* | *Description* |
| id | Integer | O | Identifiant auto généré en DB |
| idServer | String | O | Identifiant en DB: Timestamp en nanoseconde |
| active | boolean | O | Détermine si l’objet est actif ou non |
| deleted | boolean | O | Détermine si l’objet a été supprimé ou non |
| createdAt | Timestamp | O | La date de création de l’objet |
| updatedAt | Timestamp | N | La date de dernière modification de l’objet |

## Règles de transformation des données

*Décrire donnée par donnée les règles de transformations effectuées*

Les controllers reçoivent des objets DTO sur lesquels sont effectués les traitements de contrôle de validité et de cohérences des données avant d’être transformées en objet entité pour être stockées en base de données.

De même, les entités venant de la base de données, doivent être transformées en DTO avant d’être envoyées à l’application appelante.

La librairie Selma est utilisée pour la transformation des données

DB

Mapper (Selma)

Application

Appelante

asEntity

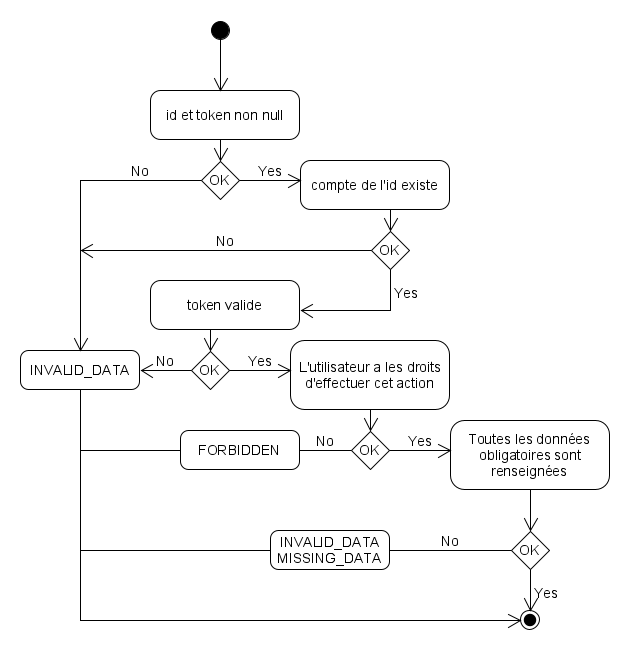
asDto

## Règles de contrôles de la donnée

*Décrire donnée par donnée les règles de contrôle effectuées*

Avant le traitement de toute requête, les règles suivantes doivent être validées

* L’identifiant du compte émetteur de la requête doit être non null et le compte doit exister en base de données
* Le token de la requête doit être valide
* Le compte, une fois identifié doit avoir les droits d’exécuter l’action demandée



## Messages d’erreur

*Décrire les messages d’erreurs (référence, libellé détaillé du message, description des mesures à prendre)*

Les messages d’erreur seront formatés de la manière suivante

|  |  |  |  |
| --- | --- | --- | --- |
| *Statut HTTP - Message global* | *Codes* | *Messages (optionnel)* | *Field (optionnel)* |
| 200 - Success |  |  |  |
| 400 - Modification impossible | INVALID\_DATA | Message pouvant aider à corriger la requête |  |
| MISSING\_DATA |  | Nom du champ concerné |
| NOT\_FOUND | L’objet n’existe pas |  |
| 400 - Erreur d’analyse de la requête | INVALID\_DATA | Message pouvant aider à corriger la requête |  |
| 403 - Modification interdite | FORBIDDEN\_ACTION | L’id de l’objet transmis ne correspond pas à celui de l’objet |  |
| FORBIDDEN\_ACTION | L’id de l’objet transmis ne correspond pas à celui de l'entête de la requête |  |
| FORBIDDEN\_ACTION | Modification du type de l’objet non autorisée |  |
| FORBIDDEN\_ACTION | Modification non autorisée |  |

## Données en sortie du service applicatif

*Décrire les données en sortie du service applicatif*

Les données en sortie sont un Object défini comme ci-dessous

|  |  |  |  |
| --- | --- | --- | --- |
| *Nom* | *Type* | *Obligatoire* | *Description* |
| status | HttpStatus | O | libellé du code http |
| message | String | O | message à titre indicatif global |
| resource | T | N | information sur l’objet |
| resources | T | N | Information sur la liste d’objets de sortie |
| log | | | Message détaillé pour comprendre l’erreur rencontrée (facultatif) |
| log.code | Enum | N | Code d’erreur |
| log.message | String | N | Message informatif pouvant aider à corriger la requête |
| log.field | String | N | Nom du champ |

Un exemple de la structure json de l’objet de sortie

[{

"status": "",

"message": "",

"resource": {

"id": ""

},

"log": [{

"code": "",

"message": "",

"field": "",

}],

}]