

J. Racaud;A. Simon;J. Harrault;J. Blondeel;S. Daguenet;F. Corradin

Music Sheet Writer

Music Sheet Writer

Bilan Architecture

Objectifs du document

Résumé

Glossaire

Description du document

|  |  |
| --- | --- |
| Titre | Music Sheet Writer : Bilan Architecture |
| Date de création | 21/12/2015 |
| Date de publication | 10/01/2016 |
| Auteur | J. Racaud;A. Simon;J. Harrault;J. Blondeel;S. Daguenet;F. Corradin |
| Responsable | Jonathan Racaud |
| E-mail | musicsheetwriter\_2017@labeip.epitech.eu |
| Sujet | Bilan Architecture |
| Version du modèle | 2.0 |

Tableau des révisions

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Auteur | Section(s) | Commentaire |
| 21/12/2015 | Jeremy HARRAULT | Toutes | Création du document à partir de la version 1.0 |
| 04/01/2016 | Jonathan Racaud | Introduction | Rédaction |

Sommaire

[1. Introduction 1](#_Toc438422828)

[1.1. Rappel de l’EIP 1](#_Toc438422829)

[1.2. Contexte et périmètre du projet 1](#_Toc438422830)

[2. Représentation de l’architecture globale 2](#_Toc438422831)

[3. Vue globale du projet 3](#_Toc438422832)

[3.1. Cas d’utilisation principaux 3](#_Toc438422833)

[3.2. Cas d’utilisation détaillés 3](#_Toc438422834)

[4. Vue logique de l’application 4](#_Toc438422835)

[4.1. Le logiciel 4](#_Toc438422836)

[4.2. Le site internet 4](#_Toc438422837)

[4.3. Les applications mobiles 4](#_Toc438422838)

[5. Vue processus 5](#_Toc438422839)

[5.1. Le logiciel 5](#_Toc438422840)

[5.2. L’API 5](#_Toc438422841)

[5.3. Les applications mobiles 5](#_Toc438422842)

[6. Vue déploiement 6](#_Toc438422843)

[6.1. Le logiciel 6](#_Toc438422844)

[6.2. Le site internet & les applications mobiles 6](#_Toc438422845)

[7. Implémentation 7](#_Toc438422846)

[7.1. Le logiciel 7](#_Toc438422847)

[7.2. L’API 7](#_Toc438422848)

[7.3. Les applications mobiles – iPhone 7](#_Toc438422849)

[7.4. Les applications mobiles – Android 7](#_Toc438422850)

[7.5. Les applications mobiles – Windows Phone 7](#_Toc438422851)

[8. Vue données 8](#_Toc438422852)

[8.1. Modélisation 8](#_Toc438422853)

[8.2. Description 8](#_Toc438422854)

[9. Taille et performance 9](#_Toc438422855)

[10. Annexes 10](#_Toc438422856)

[10.1. Le document du CDC 10](#_Toc438422857)

Liste des Tableaux

**No table of figures entries found.**

Liste des Figures

**No table of figures entries found.**

# Introduction

## Rappel de l’EIP

EPITECH est l’école de l’expertise informatique, transformant une passion en véritable expertise. L’apprentissage à EPITECH est fondé sur une pédagogie par projets, individuels ou en groupe, validant un certain nombre de connaissances et de notions à assimiler. Tout au long de leur cursus, les étudiants se familiarisent avec le milieu professionnel, notamment grâce aux stages en première, troisième et cinquième année d’une période de quatre à six mois. L’école forme les étudiants à s’adapter à des situations inhabituelles avec la mise en place de rush (projets à réaliser sur un weekend, sur des sujets et notions dont les élèves n’ont aucune connaissance) ou le départ à l’international pendant leur quatrième année ; année durant laquelle l’étudiant va devoir faire preuve d’autonomie et de capacité d’adaptation.

Les Epitech Innovative Projects sont des projets à réaliser sur le cycle master du cursus Epitech. Ils sont conçus à la manière d’un véritable projet entrepreneurial, dans toutes ses composantes : business, techno, design & communication. Un EIP est appelé à devenir une start-up viable. Le but de l’EIP est donc de faire découvrir aux étudiants le monde de l’entreprenariat en leur demandant de mettre un place un projet et de le réaliser en faisant face à des difficultés qu’ils n’avaient jusqu’alors pas rencontrées. Le principal obstacle est la gestion de groupe composé de membres dispersés dans des pays différents, faisant face alors aux problèmes de gestion du temps et des zones horaires pour leur quatrième année. Les problématiques de communication et de vente du produit sont aussi abordées.

## Contexte et périmètre du projet

Music Sheet Writer est un logiciel d’édition de partition destiné aux musiciens néophytes et amateurs qui n’ont pas forcément les connaissances théoriques du solfège pour écrire leurs compositions. Il se présente donc comme tout logiciel d’édition de partition existant, mais apporte une fonctionnalité majeur : la génération d’une partition depuis un piano ou une guitare branchés à l’aide d’un câble JACK ou d’une interface audio USB.

Le mot d’ordre de Music Sheet Writer est d’être simple d’utilisation. En effet, en ajoutant cette fonctionnalité, nous simplifions la phase d’écriture lors de la composition d’une musique. Laissant l’utilisateur se concentrer sur la musique avant son écriture. Bien entendu, les musiciens aguerris ne seront pas en reste puisque Music Sheet Writer incorporera les outils qui leur permettront d’écrire leurs musiques de manière très précise.

Ce projet est composé de plusieurs projets. Le principal est le logiciel d’édition de partition à proprement parler. Il sera disponible à la fois sur les plateformes Windows et Mac OS. Le second projet est le site internet qui permettra de vendre le logiciel. Ce site internet aura aussi des fonctionnalités communautaires basiques (partage de partition, suivi d’utilisateurs, mise en favoris de partitions…). Enfin, trois applications mobiles seront aussi développées et elles permettront d’accéder à l’aspect communautaire développé autour de Music Sheet Writer.

# Représentation de l’architecture globale

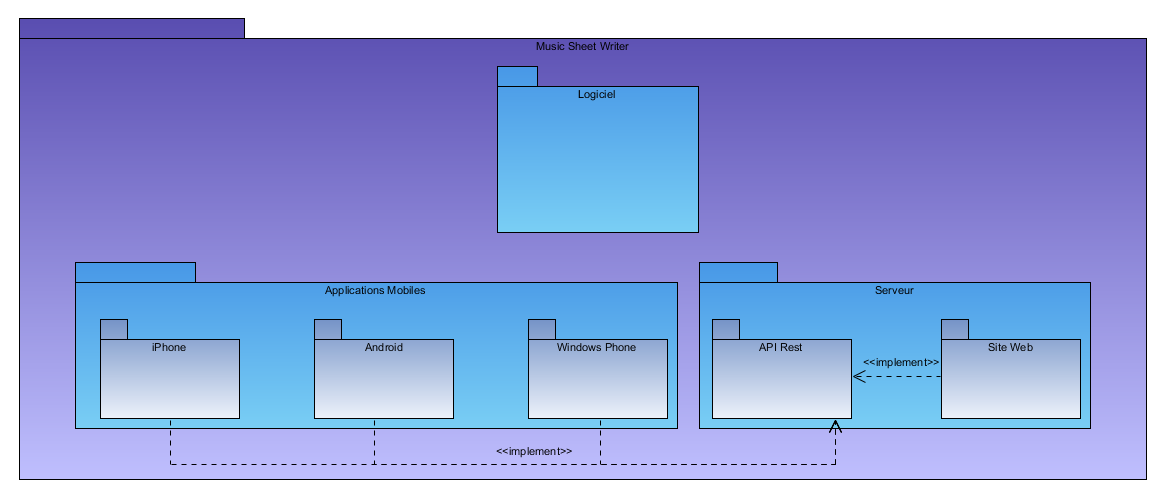


Figure 1: Architecture globale

Le logiciel est le principal livrable du projet Music Sheet Writer. Il est aussi le seul qui ne dépend pas des autres projets pour sa réalisation.

Le site web est découpé en deux projets. Le premier étant le site vitrine avec un espace communautaire. Le second est le développement d’une API Rest pour cet espace communautaire.

Cette API est utilisée par les applications mobiles, qui donnent les mêmes fonctionnalités que l’espace communautaire du site internet.

# Vue globale du projet

## Cas d’utilisation principaux

### Le logiciel

### Le site internet

### Les applications mobiles

## Cas d’utilisation détaillés

### Le logiciel

#### Editer une partition

#### Lire une partition

#### Gérer un projet

### Le site internet & application mobile

#### Accéder à la communauté

#### Gérer son compte utilisateur

#### Accéder à la vitrine (site internet uniquement)

# Vue logique de l’application

Pour chacun des livrables, la vue logique est présentée grâce à un diagramme de composant ainsi qu’une description textuelle mettant en avant les composants essentiels du diagramme.

## Le logiciel

## Le site internet

## Les applications mobiles

# Vue processus

## Le logiciel

## L’API

### Création de compte

#### Description

#### Vue dynamique

### Connexion

#### Description

#### Vue dynamique

### Changement d’adresse email

#### Description

#### Vue dynamique

### Réinitialisation du mot de passe

## Les applications mobiles

# Vue déploiement

## Le logiciel

## Le site internet & les applications mobiles

# Implémentation

## Le logiciel

### Vue globale

### Couches applicatives

## Le site internet

### Vue globale

### Couches applicatives

#### API Bundle

Le bundle APIBundle est le bundle qui implémente l’API Rest mise à disposition des applications mobiles et de l’espace communautaire du site internet.

##### Component

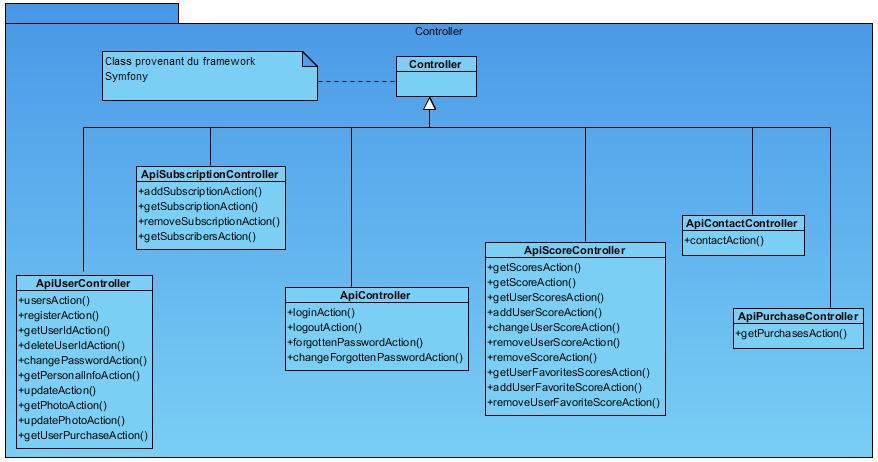
Les components sont des classes utiles pouvant être réutilisée par n’importe qu’elle autre classe présent dans l’APIBundle.

La classe MSWResponse hérite de la classe Response donné par le framework Symfony. Cette classe est utilisée par tous les Controller (un Controller doit forcément retourner une réponse).

La classe MSWSerializer nous permet de sérialiser toutes les données au format JSON (format utilisé dans la définition de l’API) ou de retrouver les différents objets envoyé au format JSON par les différents clients de l’API.

La classe MSWUtilities regroupe les différentes méthodes ou attributs qui sont utiles au développement de l’API.

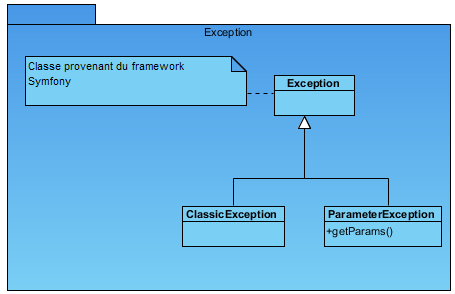
##### Controller



Les Controller héritent tous de la classe Controller donnée par le framework Symfony. Ce sont eux qui implémentent les différentes routes de l’API Rest.

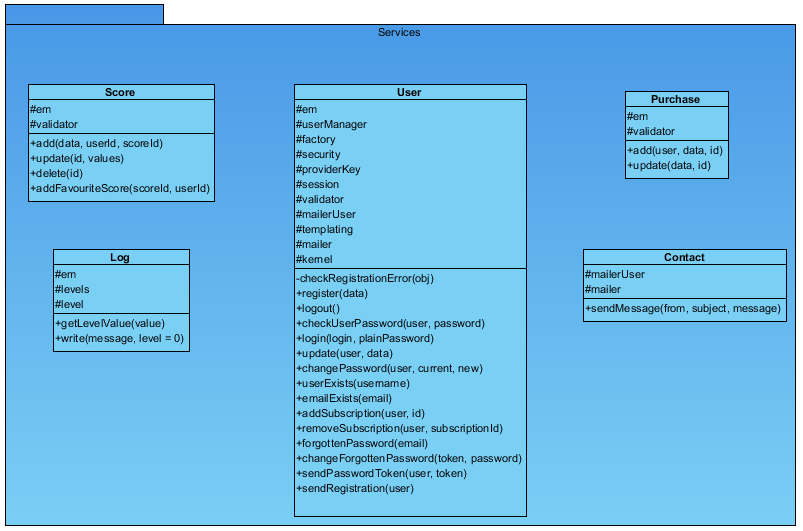
Les routes ont toutes été regroupée par domaine : les routes concernant les utilisateurs se retrouvent dans le Controller APIUserController, celles concernant les partitions dans APIScoreController, etc.

##### Exception



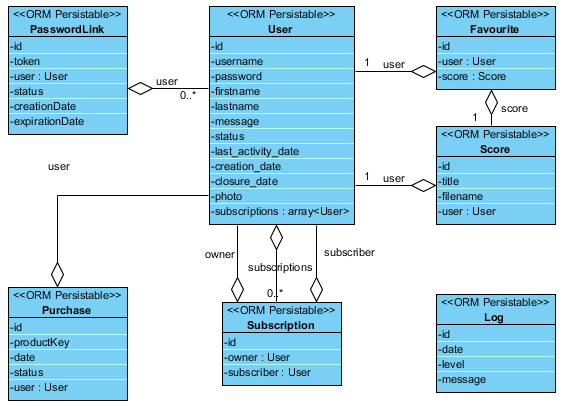
Nous avons définis nos propres exceptions, basées sur la classe Exception de Symfony. Elles sont principalement utilisées par les différents Services que nous avons définis pour implémenter le comportement de l’API Rest.

##### Services



Les services sont des objets permettant la réalisation d’opérations spécifiques sur un ou plusieurs objets et qui peuvent être réutilisé n’importe où dans le code. Par exemple, le service User permet de persister une entité User en base de donnée. Cette action est redondante et peux s’effectué à différents endroits dans le code.

##### Entity

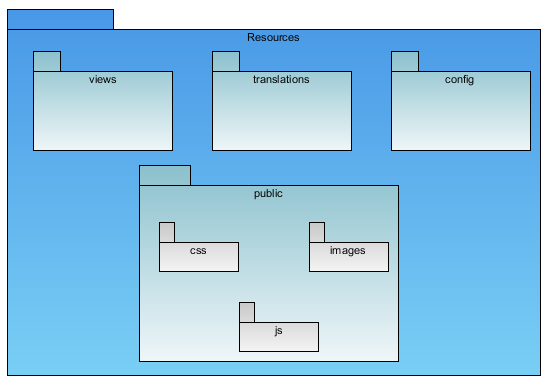


Voici les différentes entités utilisé par l’API Rest. Ces entités sont aussi le schéma de définition de la base de données utilisé par l’API (cf. : [8.1 Vue données](#_Vue_données)).

#### MSW Bundle

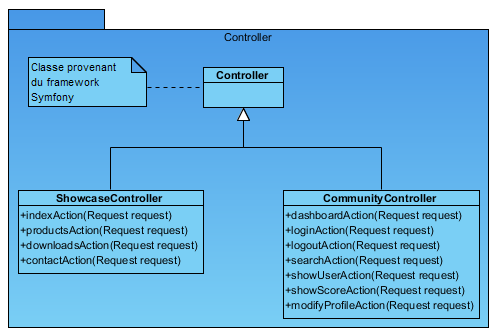
Le bundle MSWBundle est le bundle réalisant le site internet du projet Music Sheet Writer. Ce dernier étant client de l’API Rest précédemment définie, il ne contient aucune entité spécifique.

##### Resources



Le package Resources regroupe les différentes ressources utilisées par le bundle MSWBundle pour la réalisation du site internet. Il s’agit des différentes vues (views ou template twig), des fichiers de traduction (translations), de configuration (config) ou encore tous les fichiers nécessaires à un site internet (public).

##### Controller



Tout comme les controllers du bundle APIBundle qui implémentent le comportement de l’API Rest, les controllers du MSWBundle permettent d’associer les différentes vues aux différentes routes du site internet.

## Les applications mobiles – iPhone

### Vue globale

### Couches applicatives

## Les applications mobiles – Android

### Vue globale

### Couches applicatives

## Les applications mobiles – Windows Phone

### Vue globale

### Couches applicatives

# Vue données

## Modélisation

## Description

# Taille et performance

# Annexes

CTRL clic permet de visualiser ce(s) document(s)

## Le document du CDC

