

User Guidelines for 3rd party Applications and Plugins development

Introduction

The aim of this document is to provide developers of applications through basic concepts of PLUGGY. More information about PLUGGY platform could be explored on official page:

<http://pluggy-project.eu/>

PLUGGY as a place for sharing and bringing culture closer to everyone allows developers to create enhanced and non-traditional experience for all the users via exploring content with their applications. As 3rd party developer you can create two types of applications:

- Plugin integrated within PLUGGY social platform
- Standalone desktop or mobile or web application connected to PLUGGY back-end

These applications will be available to wide audience and promoted via PLUGGY. Every registered users on PLUGGY social platform (<https://beta.pluggy.eu/>) is able to apply for developers account that will bring more information and examples how to start development. For those that want to jump directly into development please visit this page for more information about API:

<https://beta.pluggy.eu/api>

Probably the most important document is API specification:

<https://beta.pluggy.eu/api/doc/>

Next few sections will try to explain you how to use PLUGGY back-end and how to create new experience for the users interested in culture or even better you can make culture interesting to people that are not into it and also you will find information how to make your application available via PLUGGY social platform to wide audience.

Stay tuned to PLUGGY and have eyes wide open to culture.

Disclaimer #1: Please keep in mind that PLUGGY is still in development process.

Disclaimer #2: This document is early developers guide, some information can be updated on the fly when necessary.

Examples for inspiration

PLUGGY brings 4 different applications that user can be inspired by. Gaming application (*link will be provided soon*), 3D sonic application (*link will be provided soon*), Geoloaction application (*link will be provided soon*) and Augmented reality application (*link will be provided soon*).

You can also visit our repository with simple examples where you can find simple examples that can explain you basic concepts behind the PLUGGY platform:

<https://isense-gitlab.iccs.gr/PLUGGY/pluggy-examples>

Individual examples are described at the end of this document under the section Use Cases.

Overall architecture

Figure 1 shows that the PLUGGY back-end provides REST API to access all the important functionality for the applications (green boxes the figure):

- Social Platform
- Curatorial Tools
- Applications (AR, Geo, 3D)
- Games
- 3rd party applications - that the applications that you are planning to develop

From the architecture you can see that the common back-end for the applications allows to create content within your application and then it can be viewed within different one or within social platform. The Web Application Platform is a modular platform that allows usage of common functionality, like logging and security, for the Social Platform, Curatorial Tool, Application Developer Tool and as previously mentioned also for external plugins. Red boxes on the figure are called the “Core PLUGGY Components” which can be split to two main modules:

- Content Management Services
- PLUGGY Content Repository

Content Management Services will contain all needed software components to process, store and serve information to the external user tools via the REST API. These components are:

- External Repository Content Connectors - you can access by unified interface different external libraries like europeana.eu, wikipedia.org, britishmuseum.org
- Hinting/Suggestions and Recommendation Services
- Search Services - basic and advanced content search - text, geolocation, metadata, etc.
- Authorization, Authentication and IPR Services
- Social Platform Services - services used for social interaction between users and users' content
- Notification Services - user notification when new content created, based on user preferences
- Content Services - CRUD operation for PLUGGY content

All of these components access the PLUGGY Content Repositories which store all the information. More about data model and basic PLUGGY concepts are in described in next section. More information about REST API can be observed and tested within the online API documentation:

<https://beta.pluggy.eu/api/doc/>

Yellow boxes present two basic principles of extensibility of PLUGGY platform:

- Pluggable Application Interface - allow users to explore and create content of different types, that core PLUGGY application does not support
- External Repositories Content Connectors - allow users to reuse content from existing external content repositories

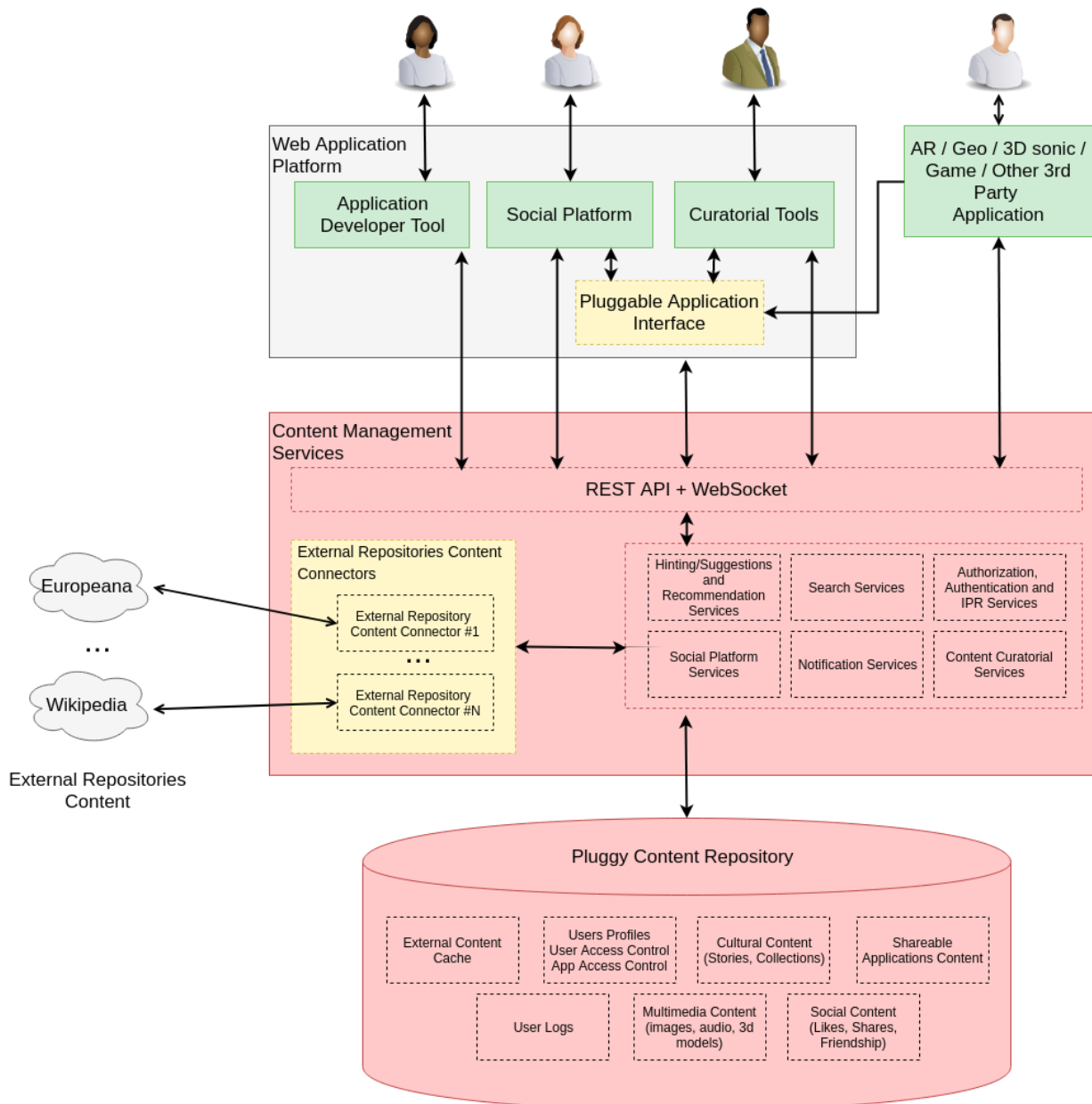


Figure 1: PLUGGY Architecture

PLUGGY's Data Model

Content is the most important part of the platform, users are interested in it. They would like to explore it virtually or even better in real with help of IT. The basic content concepts in PLUGGY are:

- Asset
- Exhibition

Asset (<https://beta.pluggy.eu/api/doc/#/assets>) is elementary unit that is digital representation of cultural heritage objects (text, image, sound, 3d model, any type of binary data). Every asset has clearly defined owner and license who can and how can be reused.

PLUGGY platform provide basic CRUD methods to manipulate with them and some advance functionality for social interaction, importing from external sources, etc.

Exhibition (<https://beta.pluggy.eu/api/doc/#/exhibitions>) basically it is way how to explore culture stories. PLUGGY provides you few basic curatorial tool for creation of exhibition like media story, timeline story, geolocation story, etc. But you are welcomed to create your own curatorial tool or viewer that engage users more.

PLUGGY platform provide basic CRUD methods for manipulation with the Exhibitions and some advance functionality for social interaction, following and notification functionalities.

Assets and Exhibitions are basic concepts that are presented to the users, but it is only base of PLUGGY. For more information explore Figure 2 and our online REST API documentation.

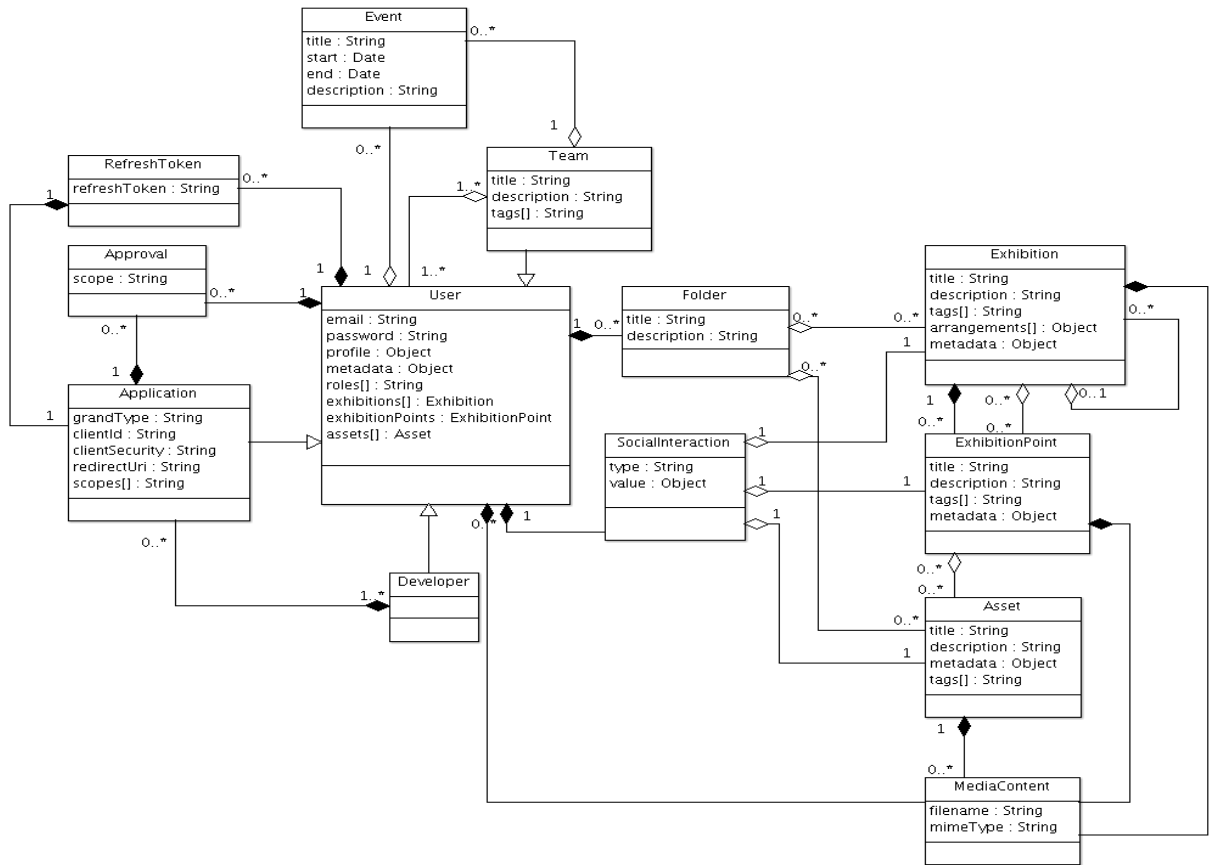


Figure 2: PLUGGY's Data Model

How to become a developer

Register yourself to PLUGGY platform and in edit section of your profile you can apply for developer's account (see Figure 3). Your request will be automatically accepted, but to apply changes you have to log out from the PLUGGY platform and then log in to it back.

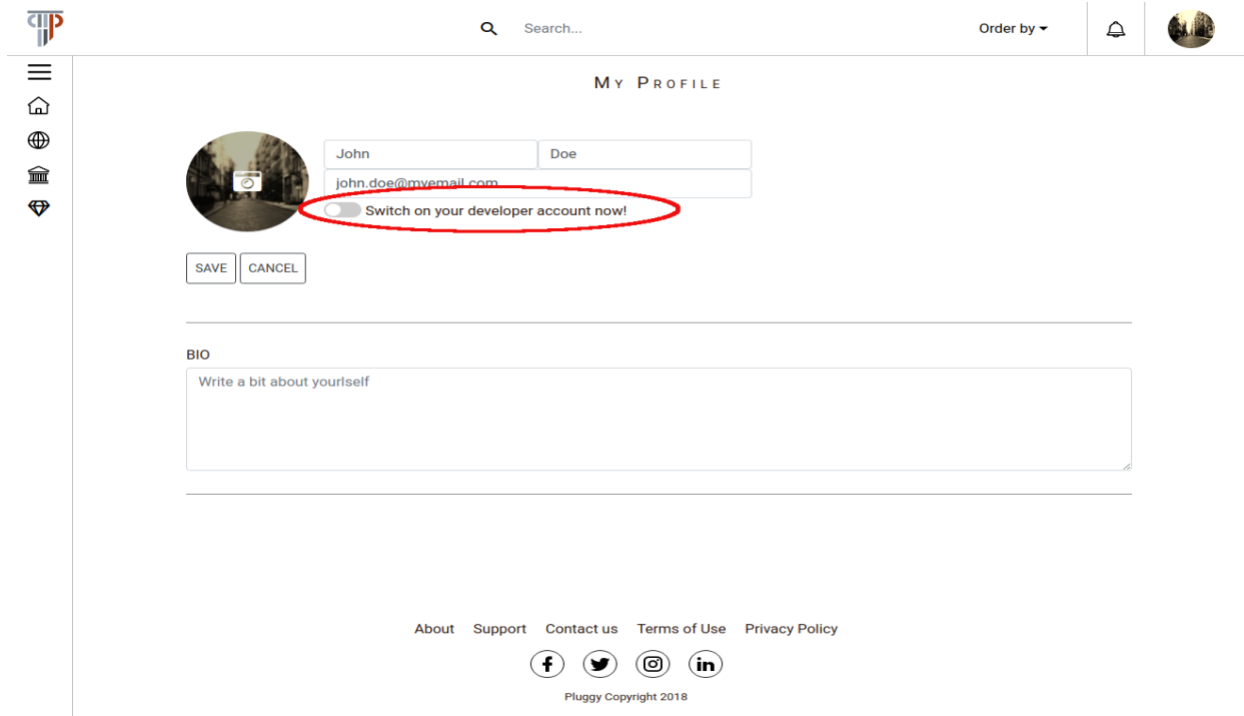


Figure 3: Applying for developer's account

Then you will be able to see applications that you are owner of or you are part of a developer team (Figure 4).

The screenshot shows the 'Applications' page on the Pluggy platform. At the top, there is a search bar and an 'Order by' dropdown. The main content is a table listing applications. The table has four columns: 'Application Name', 'URL', 'Type', and 'Trusted'. Below the table is a blue button labeled 'Create New Application'. At the bottom of the page, there are links for 'About', 'Support', 'Contact us', 'Terms of Use', and 'Privacy Policy', along with social media icons for Facebook, Twitter, Instagram, and LinkedIn. The copyright notice 'Pluggy Copyright 2018' is also present.

Application Name	URL	Type	Trusted
swagger_localhost	http://localhost:3000/api/doc/oauth2-redirect.html	Plugin	Yes
swagger_dev	https://develop.pluggy.eu/api/doc/oauth2-redirect.html	Plugin	Yes
swagger_mockup	https://mockup.pluggy.eu/api/doc/oauth2-redirect.html	Plugin	Yes
web_localhost	http://localhost:4200/auth	Plugin	Yes
web_dev	https://develop.pluggy.eu/auth	Plugin	Yes
web_mockup	https://mockup.pluggy.eu/auth	Plugin	Yes

Figure 4: List of applications that you are part of it.

Now you can register your application and connect it to pluggy platform. Which will be described in next section.

Registration of your application to the platform

Basically you can create two types of applications:

- Plugin integrated within PLUGGY social platform (see Figure 5)
- Standalone desktop or mobile or web application connected to PLUGGY back-end (see Figure 6)

It depends on developers' idea in which way would like to integrate their application to the existing PLUGGY platform. The key differences between these two types of applications are:

- Plugin is integrated into social platform and could be invoked within window of social platform.

Plugin [Stand-alone Application](#)

Plugin name

Plugin URL

Plugin Type

Trusted Plugin - This Checkbox is visible only to admin.

[Create](#)

Figure 5: Registration of the new PLUGGY plugin

Plugin Stand-alone Application

Application name

Application Type

OAuth2 Grand Type

Trusted Application - This Checkbox is visible only to admin.

Application Scope

- Public Content Viewing
- Private Content Read
- Private Content Write
- Social Interaction

Client ID

Client Security

Redirect URI

Figure 6: Registration of the new PLUGGY connected standalone application

Security

PLUGGY is a pluggable platform, multiple applications could access users' content. For this functionality we are adopting the OAuth2 (<https://tools.ietf.org/html/rfc6749>). concept. Users should enter credentials only via trusted PLUGGY's authentication interface and every application which want to access user's content has to be approved. PLUGGY will be compatible with other platforms, that support OAuth2 such as Google and Facebook. User could use their existing credentials for access to the PLUGGY platform.

Use cases 1: Creation of simplest image asset

The best practice for PLUGGY should be that one asset will contain one media file (image or audio or 3D model or ...). You can add cover image for every asset (no needed for image).

Steps to create asset:

1. create asset POST: /assets

2. (optional) upload mediaContent POST: /assets/:assetId/media
3. (optional) upload coverImage POST: /assets/:assetId/cover

When you are uploading media, cover ... You have to use "multipart/form-data" and name of file input box should be "file"!

Use cases 2: Creation of simplest sound asset

Description in progress ...

Use cases 3: Creation of simplest exhibition

You can create Exhibition - which can be of different types. PLUGGY currently supports: tour, ar, sound, game, media, timeline. Then you should create Exhibition Points with individual points/stops of exhibition.

Steps to create exhibitions:

1. create exhibition POST: /exhibitions
2. create individual exhibition points POST: /exhibitions/:exhibitionId/exhibition-points
{assets: <list of assets used for this point>, location: {geo:}, ...}

Cover image for exhibition could be uploaded: POST: /exhibitions/:exhibitionId/cover

When you are uploading media, cover ... You have to use "multipart/form-data" and name of file input box should be "file"!

Use cases 4: Search

Ex and assets

/search

Assets

/search?kind=ContentAsset

Exhibitions

/search?kind=ContentExhibition

Get all assets

/assets

Get all exhibitions

/exhibitions

Exhibitions for Assetid

/search?exhibitionsForAsset=5c120d03fffb9430d682c277

Exhibitions for Exhibition

/search?relatedExhibitions=5c120d03fffb9430d682c27f

Search by UserId

</search?user=5af19572ed0683578c1a6111>

Search in External Library - Europeana

</search?q=mummy&page=0&limit=10&sort=trending&repository=europeana>