

Projet

Application pour « le petit musée du CERI »

Objectifs

Le CERI a rassemblé quelques ordinateurs anciens et autres artefacts du passé de l'informatique, témoins du monde de l'informatique tel que l'ont connu les enseignants-chercheurs de l'établissement dans leur jeunesse, mais aussi les étudiants qui vous ont précédés. Plusieurs des objets de ce « musée » sont présentés dans la vitrine située dans le hall du bâtiment.

Dans ce projet, il vous est demandé de réaliser une application permettant d'explorer le contenu de cette collection, permettant à l'utilisateur d'accéder à la liste des objets présents et de consulter une description de chaque objet, complétée d'éventuelles images liées. Ces informations seront obtenues via l'interrogation d'un service web tournant sur un serveur du CERI.

Le projet était initialement prévu sur une durée de 3 séances de 3h, complétées d'une dernière séance consacrée à une démonstration, mais il est possible que vous ayez besoin de plus d'heures que cela pour le réaliser. Ce projet est à réaliser individuellement.

Contrairement aux sujets des TP, aucune capture d'écran n'est fournie, afin de ne pas influencer vos choix concernant la disposition, l'aspect et le fonctionnement de l'interface de l'application. Dans un premier temps, il vous est conseillé de réfléchir à votre interface graphique et votre architecture logicielle, en définissant les écrans et les classes dont vous aurez besoin. Vous pourrez passer ensuite à la réalisation des fichiers XML de mise en page et du code source.

Fonctionnalités de l'application

L'application devra présenter à l'utilisateur les informations obtenues auprès du service web consultable à l'adresse suivante :

<https://demo-lia.univ-avignon.fr/cerimuseum/>

API du service web

`/ids` → *JSON*

Tableau de chaînes de caractères représentant les identifiants de tous les objets de la collection. L'identifiant de chaque objet est unique dans la collection.

`/categories` → *JSON*

Tableau de chaînes de caractères représentant toutes les catégories d'objets figurant dans la collection.

`/catalog` → *JSON*

Dictionnaire (association clés/valeurs) contenant la collection complète. Les clés sont les identifiants des objets (les mêmes que retournés via `/ids`) et les valeurs sont les caractéristiques des objets, telles que retournées via `/items/<itemID>`.

`/items/<itemID>` → *JSON*

Dictionnaire listant les caractéristiques de l'objet spécifié (via son identifiant).

Certains champs sont systématiquement présents :

name → *chaîne de caractères*

Nom de l'objet.

categories → *tableau de chaînes de caractères*

Liste des catégories auxquelles appartient l'objet.

description → *chaîne de caractères*

Description de l'objet. Exceptionnellement, la chaîne peut être vide (dans le cas où la description n'apporterait aucune information supplémentaire par rapport aux autres champs), mais l'entrée clé/valeur sera tout de même présente dans le dictionnaire.

timeFrame → *tableau d'entiers*

Décennies durant lesquelles cet objet a été commercialisé ou utilisé.

D'autres champs ne sont présents que lorsqu'ils ont un sens pour cet objet et une valeur non vide :

year → *entier*

Année d'apparition de l'objet (généralement, il s'agit de la première année de commercialisation).

brand → *chaîne de caractères*

Marque de la société ayant produit l'objet.

technicalDetails → *tableau de chaînes de caractères*

Liste de caractéristiques techniques.

working → *booléen*

Indique si l'appareil est en état de fonctionner. L'absence de ce champ équivaut à une valeur fausse.

pictures → *dictionnaire*

Liste d'images liées à cet objet. Pour chaque entrée, la clé est l'identifiant de l'image (permettant de la récupérer via la requête idoine), et la valeur est l'éventuelle légende correspondante (souvent une chaîne vide).

`/items/<itemID>/thumbnail` → *PNG*

Image PNG de petite taille représentant l'objet, pour affichage dans une liste ou dans une présentation détaillée de l'objet.

`/items/<itemID>/images/<imageID>` → *JPEG*

Image JPEG de grande taille illustrant l'objet.

`/demos` → *JSON*

Dictionnaire contenant les dates (et heures) des 10 prochaines séances (fictives) de démonstration de certains des appareils en état de fonctionner.

Les clés sont des identifiants d'objets. Les valeurs sont des chaînes de caractères représentant des dates au format ISO 8601 (dans le fuseau horaire UTC).

Fonctionnalités à assurer

Affichage du catalogue :

L'application devra afficher la liste des objets présents dans la collection, faisant apparaître pour chacun son image en miniature, son nom (et, le cas échéant, sa marque), ainsi que les catégories auxquelles il appartient.

Cette liste devra pouvoir être triée par ordre alphabétique ou par ordre chronologique.

Affichage par catégories :

Un troisième mode d'affichage du catalogue devra présenter la liste des objets découpée en sections correspondant aux diverses catégories d'objets. Chaque section affichera en titre le nom de la catégorie, puis la liste des objets appartenant à cette catégorie.

Recherche :

Un champ de recherche devra être présent pour permettre de filtrer le catalogue via une requête textuelle simple fouillant dans toutes les caractéristiques des objets.

Vue détaillée d'un objet :

La sélection d'une entrée dans les vues précédentes amènera à une vue dédiée à la présentation détaillée de l'objet correspondant, affichant toutes les informations disponibles au sujet de cet objet.

Le choix du style de présentation est laissé libre, en particulier concernant l'affichage des images liées à l'objet.

Optionnel : amélioration de l'affichage par ordre chronologique

Le but est de faire mieux qu'un simple tri par ordre chronologique de la liste des objets. Deux approches vous sont proposées.

Version simple : Lors de l'affichage par ordre chronologique, découper la liste en sections correspondant aux décennies. Pour chaque objet dans la liste, afficher son année de production si l'information est présente.

Version avancée : Afficher les entrées du catalogue sous forme d'une frise chronologique.

Cette partie du travail étant optionnelle, ne pas la réaliser ne vous coûtera aucun point sur la note de votre projet. Mais si vous la réalisez de façon réussie, elle vous apportera des points supplémentaires : jusqu'à 1 point pour la version simple, jusqu'à 2 points pour la version avancée.

Rendu

Une démonstration est demandée à une date qui reste encore à fixer.

Il vous est demandé de déposer votre projet Android (répertoire `src` uniquement) sur GitLab. Vous déposerez sur la page du cours e-uapv un rapport en PDF contenant un lien vers votre projet, une présentation des écrans de votre application et une explication de vos choix techniques. Votre dépôt devra également intégrer un `apk` prêt à être déployé sur un smartphone. La date limite pour la prise en compte de ces dépôts est la veille de la démonstration.

Dans la notation seront pris en compte le respect des échéances, le respect des fonctionnalités demandées et la qualité de l'interface utilisateur (l'ergonomie primant, mais l'esthétique n'étant pas à négliger pour autant).