Mobile Application Store

Collection API Design Document

Date: 10/29/2013

Author: Siddharth SINGH

Reviewer(s): Hemant BAJPAI

Introduction

This document defines design document for collection service of mobile application store.

Overview

Collection service does logical grouping of mobile application store contents. Collection service enables users to add collection and define collection parameters. It also enables users to search and iterate over search results of collection of related contents. E.g. user can search all contents in music/sports collection or user can search on basis of certain search criteria (e.g. free applications or highly rated applications).

Requirements

1. Create collection: A collection is either static or dynamic. A static collection has specific set of products whereas dynamic collection has a search criteria and it contains all the products which meet the given criteria. Create collections given collection id, name, description and type of collection (static or dynamic).
2. Add content to collection: content is either a product or another collection. Add content (product or collection) to a given collection.
3. Search collection: perform text search on collections. Return all collections which contain the text in their name or description.
4. Add search criteria to dynamic collection: Add search criteria to a given dynamic collection.
5. Iterate over collection: A collection includes list of products as well as child collections. Provide a way to iterate recursively over collection and its children.

Use Cases

There are two types of users:

Mobile Application Store Product API Design Document

CSCI E-97 Assignment 3

1. An administrator can create collection, add content to collection and search and iterate over collections.
2. A Consumer can search and iterate over collections

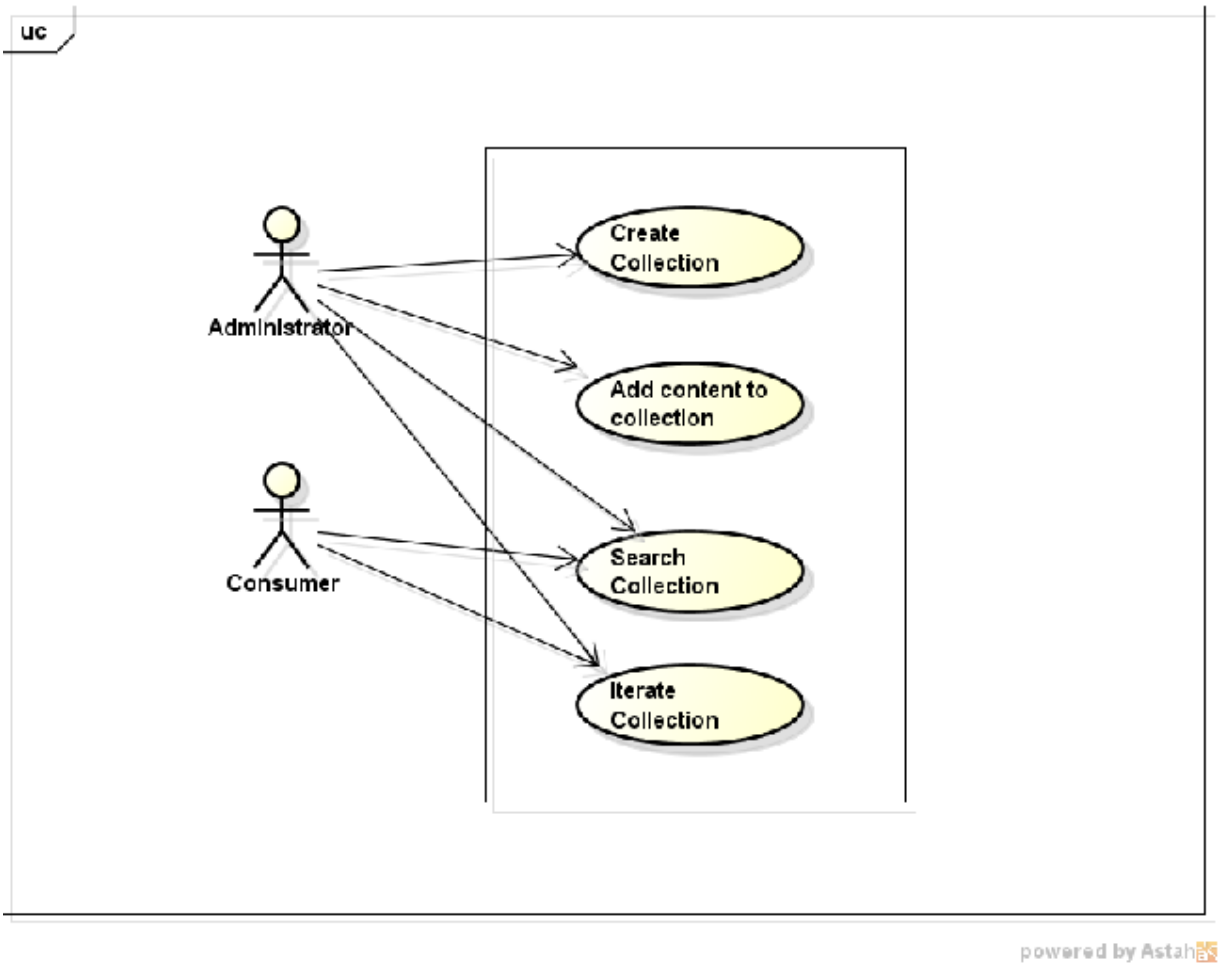
Create a collection given its id, name, type and description. Id should be unique and all the fields should be non-empty. Otherwise collection service throws exception.

Add content to collection. Given a collection add content to the collection as its children. Both collection and content should exist in the store. Otherwise collection service throws exception.

Set search criteria of dynamic collection. Given a collection id set the search criteria. Id should refer to a dynamic collection in the store otherwise throws exception.

Search collection: perform text search on collections. Return all collections which contain the text in their name or description.

Iterate over collection: given a collection id iterate over the collection and all its children. If Id is blank or null then iterate over all the top level collections in their children in the store.



Mobile Application Store Product API Design Document

CSCI E-97 Assignment 3

Implementation

A collection has set of products as well as child collections of same type. This set up is an ideal candidate for **Composite** design. Collectible is an abstract class which represents the elements which can be contained in a collection. In that sense collection and product are collectibles and a collection class composes set of collectibles. The composite pattern makes it easy to write recursive algorithms (depth first search) when it comes to iterate over its elements.

Iterator pattern enables iteration over elements of a collection. A collection iterator is associated with each collection class. Collection iterator maintains a list of child products by getting all the contained products and recursively retrieving contained products of child collections. Depth first search flattens the tree structures of collection hierarchy into a list. The iterator iterates over the flattened list by moving to next index in the list for each iteration of the list.

A collection service **factory** class defines a static method which returns the singleton collection service object.

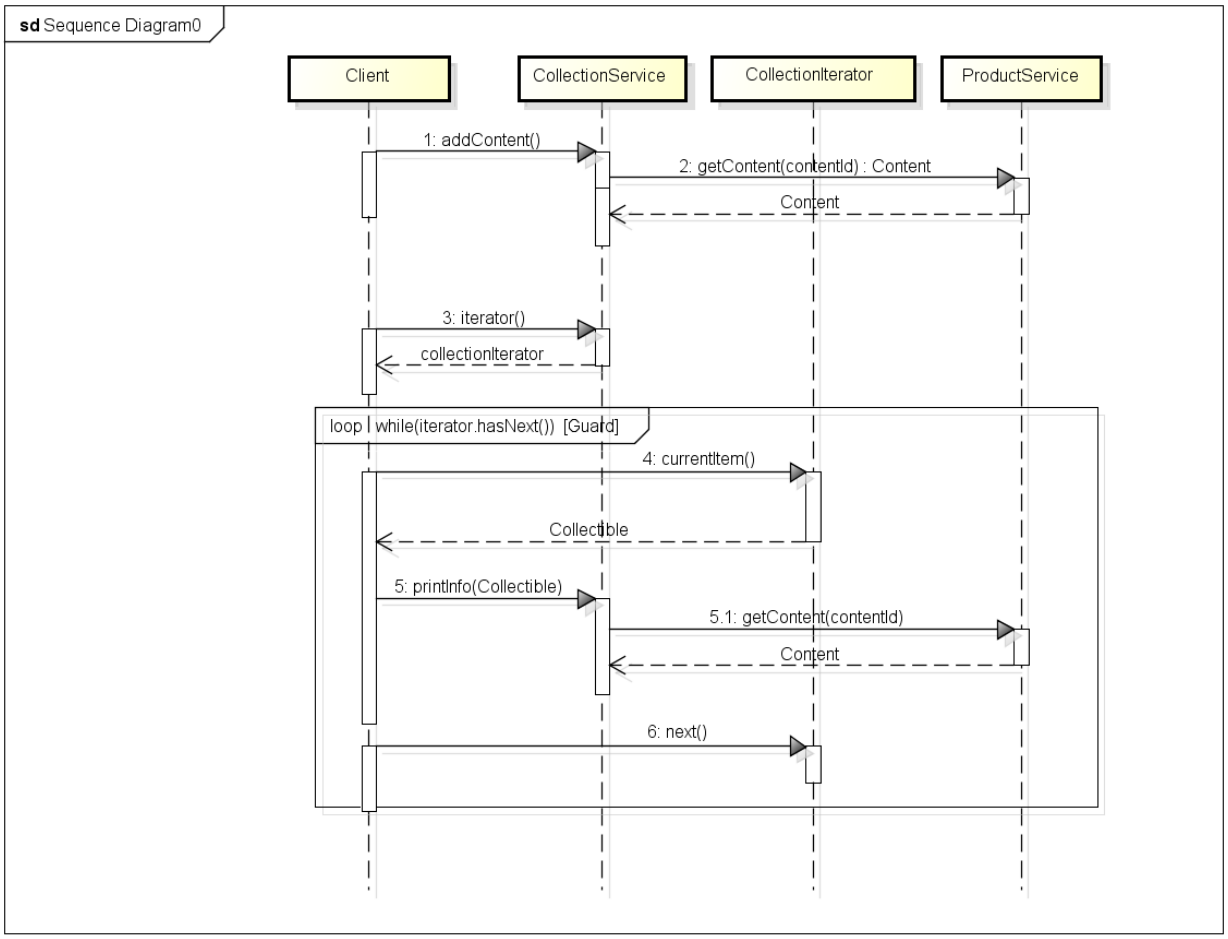
Sequence Diagram

Please see CollectionService_sequenceDiagram.png or CollectionService_classDiagram.asta for clear class diagram.

1. Client calls addContent() on collection service. If content is a product then collection service checks to see if it exists in product service. Product service return a content object if exists otherwise return null.
2. Client calls iterator on collection service and passes collection id. Collection service returns iterator for the collection tree rooted at collection with the given id.
3. While loop checks if iterator has more elements to iterate by calling hasnext() on iterator.
4. Gets current Item by calling currentItem() on iterator which returns a collectible object.
5. Client call printInfo() on collectible object which get called polymorphically depending on type of object collectible is pointing to (static collection, dynamic collection or product proxy).
6. Client goes to next item in iterator by calling next().

Mobile Application Store Product API Design Document

CSCI E-97 Assignment 3

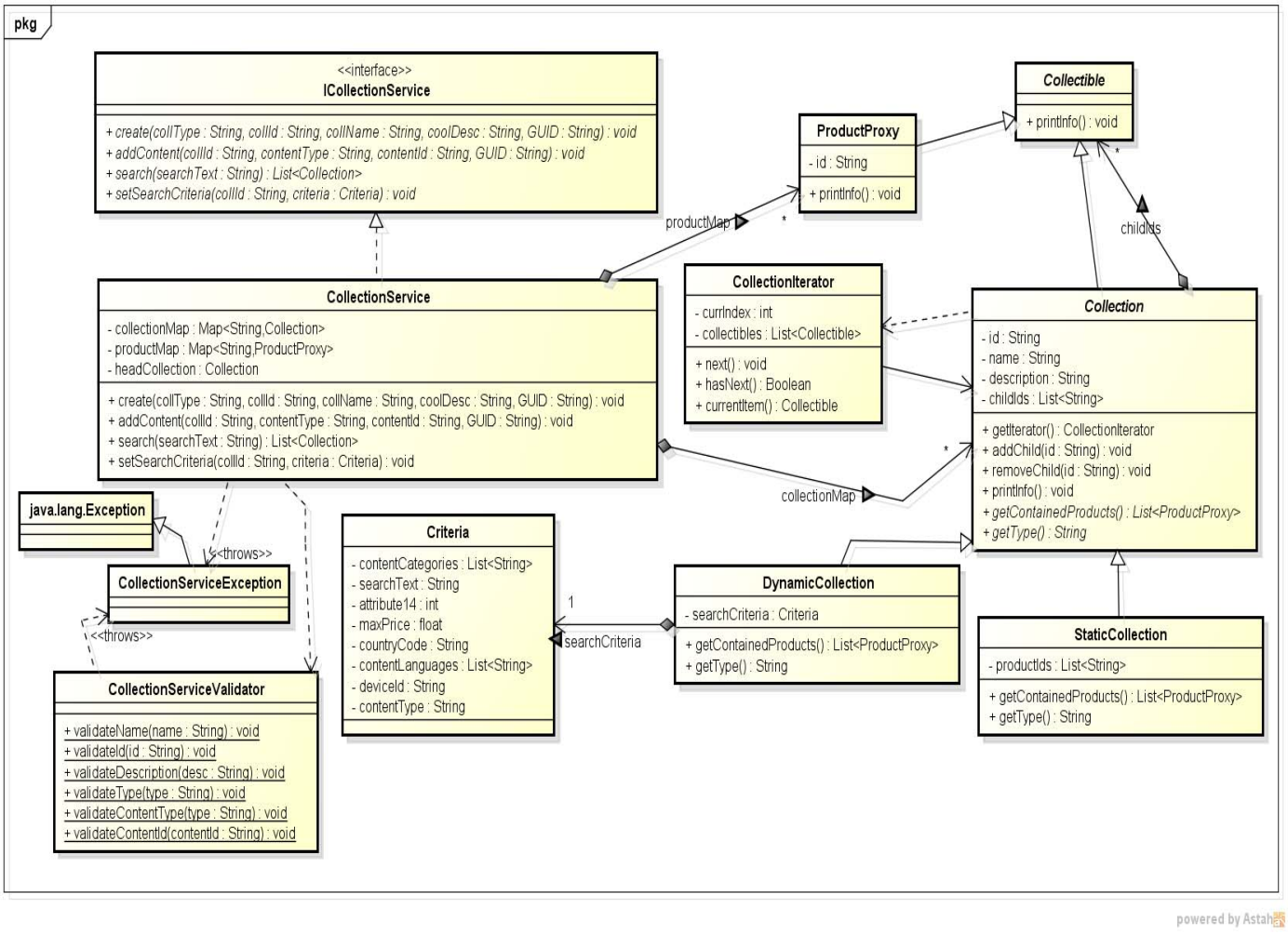


Class Diagram

Please see CollectionService_classDiagram.png or CollectionService_classDiagram.asta for clear class diagram.

Mobile Application Store Product API Design Document

CSCI E-97 Assignment 3



Class Dictionary

ICollectionService<<interface>>

Methods

Method Name	Signature	Description
create	void create(String collType,String collId,String collName,String collDesc,String GUID) throws CollectionServiceException	Creates collection with given type, id, name and description. Throws CollectionServiceException if id is not unique or type nor static nor dynamic or any of the fields are

		blank.
addContent	void addContent(String collId,String contentType,String contentId, String GUID)throws CollectionServiceException;	Add content to a given collection. Throws CollectionServiceException if collection or content does not exist in the store or any of the fields are null or blank.
search	void search(String searchText)	Perform text search on collections. Return all collections which contain the text in their name or description
iterator	CollectionIterator iterator(String collId)throws CollectionServiceException	Returns the begin iterator of the collection. If collId is blank or null then it returns a parent iterator of all the top level collection.
setSearchCriteria	void setSearchCriteria(String collId,Criteria criteria)throws CollectionServiceException;	Set search criteria for a dynamic collection. Throws CollectionServiceException if collId is blank or does not refer to a dynamic collection.

CollectionService

CollectionService implements all the methods declared in ICollectionService. It keeps map of collection id and collection object. It also keeps map of product id and product object as well as parent collection object of all the top level collection objects. This parent object acts as root of the collection tree.

Associations

Association Name	Type	Description
collectionMap	Map<String,Collection>	map of collection id and collection object
productMap	Map<String,ProductProxy>	map of product id and product object
headCollection	Collection	root of the collection tree

Collectible {abstract}

Collectible class represents elements that can be contained by a collection. Collectible is inherited by product and collection. It declares an abstract method to print the properties of collectible objects to standard output. This auxiliary method is useful for test driver.

Mobile Application Store Product API Design Document

CSCI E-97 Assignment 3

Methods

Method Name	Signature	Description
printInfo	void printInfo()	print the properties of collectible objects to standard output.

ProductProxy

This acts a proxy class for content in product api service. This class derives from collectible and implements the printInfo() method. It keeps an id of the product it represents.

Properties

Property Name	Type	Description
id	String	Id of product it represents.

Collection {abstract}

Collection class represents the composite collection which includes set of products and set collections. This is abstract because a collection does not fully qualify the method by which it can store products. So it declares an abstract method getContainedProducts() which is overridden by concrete classes DynamicCollection and StaticCollection. Otherwise this class defined all the methods and properties of a collection.

Associations

Association Name	Type	Description
childIds	List<String>	List of collectible Ids

Properties

Property Name	Type	Description
id	String	Id of collection.
Name	String	Name of collection.
description	String	Description of collection.

Methods

Mobile Application Store Product API Design Document

CSCI E-97 Assignment 3

Method Name	Signature	Description
getContainedProducts	List<ProductProxy> getContainedProducts()	Abstract method for subclasses to implement. Concrete classes know the policy of fetching products it contains. E.g. Dynamic collection calls product service API to get all products which meets its criteria.
getIterator	CollectionIterator getIterator()	Returns the begin iterator of collection object.
addChild	void addChild(String id)	Adds a child collection to the list of child collections.
removeChild	void removeChild(String id)	Removes a child collection to the list of child collections.
getType	String getType()	Abstract class for subclasses to override. Subclasses return type of collection they are (static or dynamic). Useful in printing info about the collection.

StaticCollection

Derives from Collection and override getContainedProducts(). This class keeps static list of products.

Properties

Property Name	Type	Description
productIds	List<String>	Static list of contained products.

Methods

Method Name	Signature	Description
getContainedProducts	List<ProductProxy> getContainedProducts()	Returns the static list of contained products.
getType	String getType()	Returns “static-collection”

DynamicCollection

Derives from Collection and override getContainedProducts(). At runtime it calls product service api to get all the product which meets its search criteria.

Mobile Application Store Product API Design Document

CSCI E-97 Assignment 3

Associations

Association Name	Type	Description
searchCriteria	Criteria	Search criteria of dynamic collection.

Methods

Method Name	Signature	Description
getContainedProducts	List<ProductProxy> getContainedProducts()	Calls product service API to get all products which meets its criteria
getType	String getType()	Returns “dynamic-collection”

Criteria

Defines search criteria for dynamic collection.

Properties

Property Name	Type	Description
contentCategories	List<String>	Product categories.
searchText	String	Product search text.
minRating	Int	Product rating.
maxPrice	float	Price of product.
countryCode	String	Country where product is available.
contentLanguages	List<String>	Languages product supports.
deviceId	String	Device product supports.
contentType	List<String>	Product types.

CollectionIterator

Iterator class for iterating collections.

Properties

Property Name	Type	Description
currIndex	int	Current index of the flattened list in the iterator.
visited	Map<Collection,Boolean>	Keep track of visited nodes in depth first search of iterating collection tree.

Mobile Application Store Product API Design Document

CSCI E-97 Assignment 3

Associations

Association Name	Type	Description
parentColl	Collection	Parent collection object on which iterator is operating.
collectibles	List<Collectible>	List of all collectibles in collection tree rooted at parentColl.

Methods

Method Name	Signature	Description
hasNext	Boolean hasNext()	Returns true if iterator has more items to iterate otherwise false.
currentItem	Collectible currentItem()	Returns current item in the iterator.
next	Void next()	Goes to next item in the iterator.

CollectionValidator

This auxiliary class defines set of static functions to validate collection service parameters.

Methods

Method Name	Signature	Description
validateName<<static>>	void validateName(String name)	Throws CollectionServiceException if name is empty or null.
validateDescription<<static>>	void validateDescription(String desc)	Throws CollectionServiceException if description is empty or null.
validateId<<static>>	void validateId(String id)	Throws CollectionServiceException if id is empty or null.
validateType<<static>>	void validateType(String type)	Throws CollectionServiceException if type is empty or null or not equal to “static” or “dynamic”.
validateContentType<<static>>	void validateContentType(String type)	Throws CollectionServiceException if type is empty or null or not equal to “product” or

		"collection".
validateContentId<<static>>	void validateContentId(String contentId)	Throws CollectionServiceException if contentId is empty or null.

CollectionServiceException

Represents exception class for collection API service.

Implementation Details

Collection service keeps a map of collection id and collection object. The id as key is in upper case so that all searches are case insensitive. Similarly collection service has a map of product id and product object. A product proxy class represents a product object in collection service. The actual object is owned by product api service. So collection service queries product service for any product related information.

Collection service adds content under collection and before doing that it checks if content exists in the store. If not then it throws exception. If content is a product then it checks in product service to find out if product exists in the product service of the store.

In order to do search collection service queries in the map it is keep storing collection object. For each collection object it checks if search text is substring of name or description of collection object. Search returns list of all matching collection objects.

Collection class follows composite pattern. i.e. It composes a set of collection objects apart from some satellite data. This makes the object of collection an ideal candidate for recursion and depth first search for traversing through the nodes of the collection tree. Collection iterator associated with a collection class does exactly that and maintains a list of contained collectibles as a flattened list by traversing over the tree in depth first search manner. The iterator class provides hasnext(), currentItem() and next() as its interface in order to iterate through the list.

Testing

Test Driver class calls ICollectionService.create() to create collections. It parses a csv file to get parameters of ICollectionService.create().

It parses csv file to add contents to collection. Calls ICollectionService.addContent() to do that.

Calls ICollectionService.setSearchCriteria() to set search criteria for dynamic collection. It parses csv file to get collection id for dynamic collection and a search criteria.

Finally it calls ICollectionService.search() to search collection and calls ICollectionService.iterator() for each search result from previous call in order to iterate through collection tree. It displays collectible information on standard output while iterating through it.

Risks

Collection will need to implement more methods if client need more information about collection object. As of now collection implements printInfo() to print its properties on standard output.

Collection service decouples interface and implementation such that client code can harm the consistency of collection service. So if any changes in future will have to make sure that it does not expose objects or methods to client which can damage consistency of the program.

Collection service is also exception safe so future changes will have to guarantee exception safety as well.