Mobile Application Store Product API Design Document

Date: 2013-10-08
Author: David Killeffer <rayden7@gmail.com>
Reviewer(s): Catherine Bieber <catherine.bieber@gmail.com>

Introduction

This document defines the design for the Mobile Application Store Product API, which is a core component of the Mobile Application Store.

Overview

The Product API Service component of the Mobile Application Store is used to allow end-Consumers and Administrators the ability to interact with the products that are stored in the Mobile Application Store. The Product API Service is critical to the overall success of the Mobile Application Store, since the vast majority of revenue generated by the store will be through Consumer interactions with the product catalog, and the Service is the primary broker for this interaction.

Consumers are able to search for content across all categories (application, ringtone, and wallpaper) through a rich search interface. Consumer content searches may query for content across nearly every aspect of a content item, and present the full list of matching content items back to the consumer.

Administrators may likewise search for content similarly to Consumers, but they also have the ability to manage content by adding, editing, or deleting individual content items. Application developers may also manage content similarly to Administrators.

Requirements

This section defines the requirements for the Product API Service system component. Given that future sprints will incorporate authentication and authorization (Authentication Service) and that the Product API Service will need to interact with the Authentication Service, the design accounts for the notion of ***protecting restricted interface*** use cases, and will use a mock authentication token for now (for example, see requirements #1 and #4 in "Product API Requirements" from *MobileApplicationStoreProductAPIRequirements.pdf*).

Administrators and Application Developers that have a GUID access token should be able to add Countries, Devices, and Content to the product catalog via CSV import files loaded through the Importer. When Content items are added to the product catalog they should be

validated to include all required fields. The Importer will handle allowing items to be added to the product catalog.

Consumers may search for content across several different content criteria. The SearchEngine allows consumers, administrator, and application developers to execute content searches.

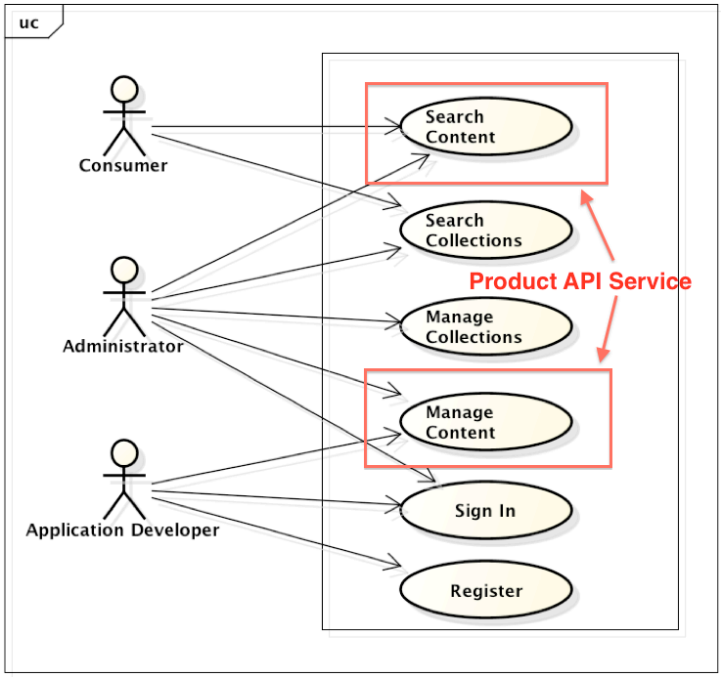
More specific details on the requirements are found in the *MobileApplicationStoreProductAPIRequirements.pdf* document.

Use Cases

The following use case diagram shows the major functional tasks that each type of user of the Mobile Application Store can perform; the use cases surrounded with red boxes denote those that the Product API Service is responsible for. In future sprint iterations the other use cases will be built out separately from the Product API Service.

This design supports the following use cases:

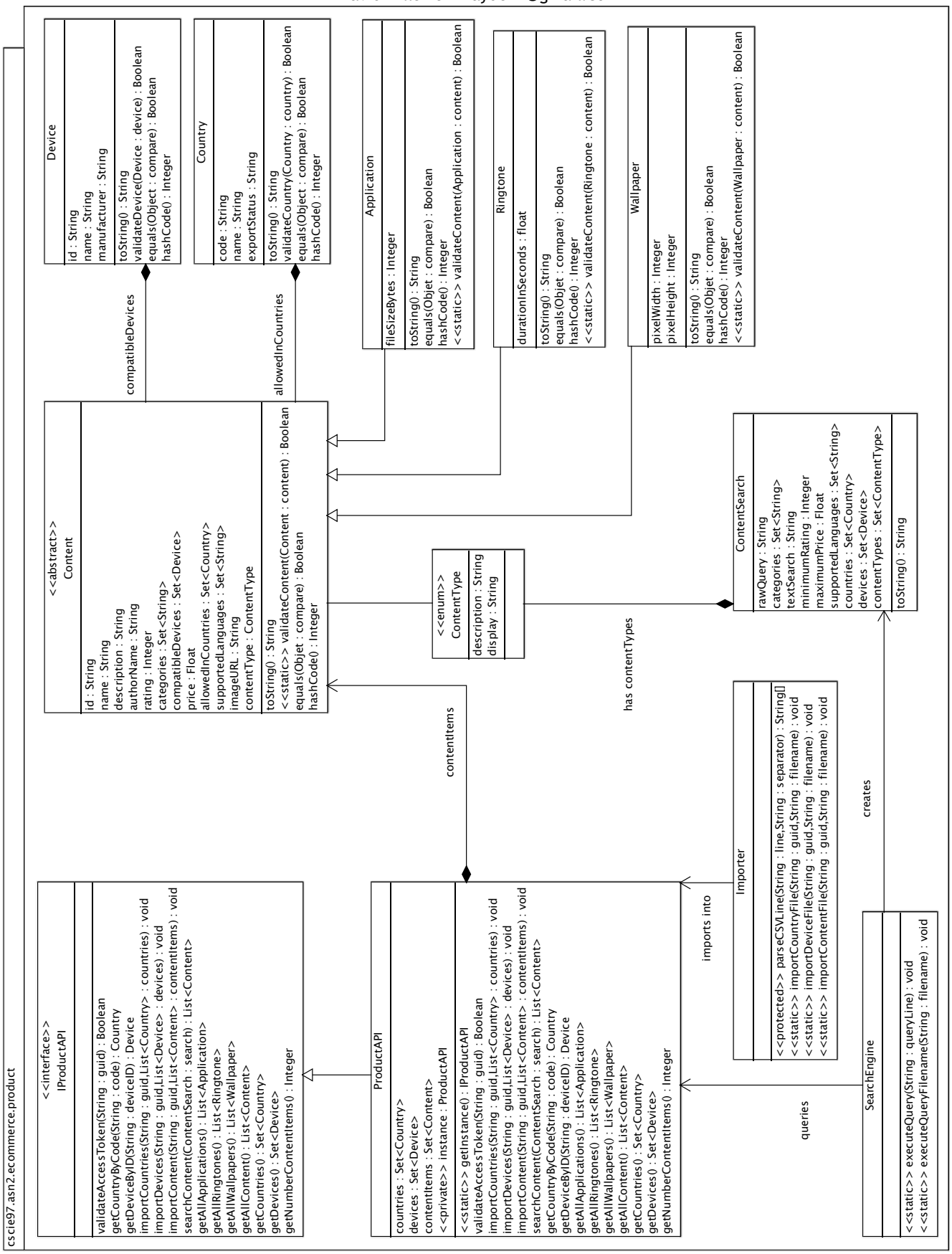
- Search Content (may be performed by Consumers or Administrators)
- Manage Content (may be performed by either Administrators or Application Developers)



Class Diagram

The following class diagram defines the classes defined in this design.

Mobile Application Store Product API Design Document
CSCI E-97 Assignment 2
David Killeffer <rayden7@gmail.com>



Class Dictionary

This section specifies the class dictionary for the Product API. The classes should be defined within the package "cscie97.asn2.ecommerce.product".

Content <<abstract>>

This abstract class provides the basic properties and methods that all content items implementing this class must follow. Its attributes are common to all content types.

Methods

Method Name	Signature	Description
validateContent <<static>>	(Content content) : Boolean	Given a content item, validate that the attributes on that content item follow the required guidelines for all content items, regardless of type. This method is called by the sub-classes of Content to ensure that all types are valid.
toString <<override>>	() : String	Overrides generic toString() method and print out all the properties of a Content item. Useful for debugging.
equals <<override>>	(Object : compare) : Boolean	Overrides the default equals method to compare the passed content item to the current one. If all properties of the two objects match completely, returns true; false otherwise. Used since Content items may be stored in the ProductAPI in collections, and objects in Sets need to be compared for equality.
hashCode	() : Integer	Overrides the default hashCode method to generate a unique code for the current object. Similar to equals(), overridden to ensure that object comparisons use the properties of the object.

Properties

Property Name	Type	Description
id	String	Unique identifier for each content item. Required.
name	String	Name for the content. Required.
description	String	A description of what the content item is. Required.
authorName	String	Name of the author of the content. Required.
rating	Integer	Content rating on scale of 0 to 5, where 5 is best. Required.
categories	Set<String>	Each unique category that the content is in. Required.
compatibleDevices	Set<Device>	All the Devices that this content item is compatible with. Required.
price	Float	Price for the content item in Bitcoins. 0 means "free".
allowedInCountries	Set<Country>	All the unique countries that the content is legal to be exported to. Required.
supportedLanguages	Set<String>	A list of the 5-character language codes that the content supports. Required.

imageURL	String	A string URL pointing to an image for the content; box-art, screenshot, etc. Required.
contentType	ContentType	Instance of the ContentType class that describes what kind of Content item the current object is. Current values are ContentType.APPLICATION, ContentType.RINGTONE, or ContentType.WALLPAPER.

Application

Models an application content item; extends Content.

Methods

Method Name	Signature	Description
validateContent <<static>>	(Content content) : Boolean	Given an Application content item, validate that the attributes follow the required guidelines for an Application content item. Calls Content.validateContent() to validate the basic properties common to all content items.
toString <<override>>	() : String	Overrides generic toString() method and print out all the properties of an Application content item. Useful for debugging.
equals <<override>>	(Object : compare) : Boolean	Overrides the default equals method to compare the passed Application content item to the current one. If all properties of the two objects match completely, returns true; false otherwise. Used since Application items may be stored in the ProductAPI in collections, and objects in Sets need to be compared for equality.
hashCode	() : Integer	Overrides the default hashCode method to generate a unique code for the current object. Similar to equals(), overridden to ensure that object comparisons use the properties of the object.

Properties

Property Name	Type	Description
fileSizeBytes	Integer	The number of bytes the application is on disk. Required.

Ringtone

Models a ringtone content item; extends Content.

Methods

Method Name	Signature	Description
validateContent <<static>>	(Content content) : Boolean	Given a Ringtone content item, validate that the attributes follow the required guidelines for a Ringtone content item. Calls Content.validateContent() to validate the basic properties common to all content items.
toString	() : String	Overrides generic toString() method and print out all the properties of a

<<override>>		Ringtone content item. Useful for debugging.
equals <<override>>	(Object : compare) : Boolean	Overrides the default equals method to compare the passed Ringtone content item to the current one. If all properties of the two objects match completely, returns true; false otherwise. Used since Ringtone items may be stored in the ProductAPI in collections, and objects in Sets need to be compared for equality.
hashCode	() : Integer	Overrides the default hashCode method to generate a unique code for the current object. Similar to equals(), overridden to ensure that object comparisons use the properties of the object.

Properties

Property Name	Type	Description
durationInSeconds	Float	How many seconds in length the ringtone is. Required.

Walpaper

Models a wallpaper content item; extends Content.

Methods

Method Name	Signature	Description
validateContent <<static>>	(Content content) : Boolean	Given a Wallpaper content item, validate that the attributes follow the required guidelines for a Wallpaper content item. Calls Content.validateContent() to validate the basic properties common to all content items.
toString <<override>>	() : String	Overrides generic toString() method and print out all the properties of a Wallpaper content item. Useful for debugging.
equals <<override>>	(Object : compare) : Boolean	Overrides the default equals method to compare the passed Wallpaper content item to the current one. If all properties of the two objects match completely, returns true; false otherwise. Used since Wallpaper items may be stored in the ProductAPI in collections, and objects in Sets need to be compared for equality.
hashCode	() : Integer	Overrides the default hashCode method to generate a unique code for the current object. Similar to equals(), overridden to ensure that object comparisons use the properties of the object.

Properties

Property Name	Type	Description
pixelWidth	Integer	How tall the wallpaper is in pixels. Required.
pixelHeight	Integer	How tall the wallpaper is in pixels. Required.

Device

Models a device that content items can be downloaded to.

Methods

Method Name	Signature	Description
validateDevice <<static>>	(Device : device) : Boolean	Given a Device, validate that the attributes follow the required guidelines for a Device. Called when importing new Devices into the product catalog to ensure that only valid Devices get added.
toString <<override>>	() : String	Overrides generic toString() method and print out all the properties of a Device. Useful for debugging.
equals <<override>>	(Object : compare) : Boolean	Overrides the default equals method to compare the passed Device to the current one. If all properties of the two objects match completely, returns true; false otherwise. Used since Device items may be stored in the ProductAPI in collections, and objects in Sets need to be compared for equality.
hashCode	() : Integer	Overrides the default hashCode method to generate a unique code for the current object. Similar to equals(), overridden to ensure that object comparisons use the properties of the object.

Properties

Property Name	Type	Description
id	String	The unique identifier for the device. Required.
name	String	Name of the device. Required.
manufacturer	String	Name of the manufacturer of the device. Required.

Country

Models a unique country where content may or may not be allowed to be downloaded to devices connected in this country.

Methods

Method Name	Signature	Description
validateCountry <<static>>	(Country : country) : Boolean	Given a Country, validate that the attributes follow the required guidelines for a Country. Called when importing new Countries into the product catalog to ensure that only valid Countries get added.
toString <<override>>	() : String	Overrides generic toString() method and print out all the properties of a Country. Useful for debugging.
equals <<override>>	(Object : compare) : Boolean	Overrides the default equals method to compare the passed Country to the current one. If all properties of the two objects match completely, returns true; false otherwise. Used since Country items may be stored in the ProductAPI in collections, and objects in Sets need to be compared for equality.
hashCode	() : Integer	Overrides the default hashCode method to generate a unique code for the

		current object. Similar to equals(), overridden to ensure that object comparisons use the properties of the object.

Properties

Property Name	Type	Description
code	String	Unique 2-character country code. Required.
name	String	Name of the country. Required.
exportStatus	String	Export status of the country. Can be either "open" or "closed". Required.

ContentType

Marker class for the unique types of discrete content supported by the Mobile Application Store. Current valid types for ContentType objects are APPLICATION, RINGTONE, and WALLPAPER. When creating an Application, Ringtone, or Wallpaper instance a ContentType matching the object's type is created and set on the content item object.

Properties

Property Name	Type	Description
description	String	1 or 2 sentence description of what the enum item is.
display	String	Short, 1-word description of what the enum item is.

ContentSearch

Captures the attributes specified in a content item search so that a single object can be passed from SearchEngine to ProductAPI, and also in the future could be more easily logged or reported on. Note that the ContentSearch does not execute any search logic itself, it merely contains the attributes to search for; the ProductAPI bears the responsibility for executing the ContentSearch by comparing its properties to everything in the product catalog.

Methods

Method Name	Signature	Description
toString <<override>>	() : String	Overrides generic toString() method and print out all the properties of a ContentSearch item. Useful for debugging.

Properties

Property Name	Type	Description
rawQuery	String	Contains the CSV line comprising the original search query sent to SearchEngine.

categories	Set<String>	Any content categories to match on any existing content items in the Product catalog on.
textSearch	String	Basic text to search across all content items for. This text will be searched across the content item name, description, and authorName atrributes of the content item.
minimumRating	Integer	The rating of content to search for on a 0 to 5 scale where 5 is best. Default value is 6 so that this parameter will not match any content items unless the search criteria specifically sets the value to something between 0-5 (inclusive).
maximumPrice	Float	The maximum price to search for content items on. Defaults to -1 so that a basic search where this value isn't set won't automatically match everything in the catalog.
supportedLanguages	Set<String>	The list of supported languages to search content items for. Note that partial string matches are supported (e.g., if a supportedLanguage code has a string "fr", and there is a content item with language code "fr_CA", the content item will match).
countries	Set<Country>	List of countries to match content items on.
Devices	Set<Device>	List of devices to match content items on.
contentTypes	Set<ContentType>	List of ContentTypes to match content items on.

Importer

Used to load Devices, Countries, and Content items (Applications, Wallpapers, and Ringtones) into the Product catalog.

Methods

Method Name	**Signature**	**Description**
parseCSVLine <<protected>>	(String : line, String : separator) : String[]	Utility method used by both Importer and SearchEngine to handle parsing CSV lines in all the import files (countries.csv, devices.csv, and products.csv). Takes a line with comma separated values and parses it into a String array.
importCountryFile <<static>>	(String : guid, String : filename) : void : throws ImportException, ParseException	Imports a CSV file and parses each line, attempting to add a new Country to the Product catalog for each line. Throws ImportException or ParseException for file handling issues or formatting problems with the CSV. *Restricted Interface - must supply a GUID token to import content [currently mocked; all strings accepted]*
importDeviceFile <<static>>	(String : guid, String : filename) : void : throws ImportException, ParseException	Imports a CSV file and parses each line, attempting to add a new Device to the Product catalog for each line. Throws ImportException or ParseException for file handling issues or formatting problems with the CSV. *Restricted Interface - must supply a GUID token to import content [currently mocked; all strings accepted]*
importContentFile	(String : guid, String :	Imports a CSV file and parses each line, attempting to add a

<<static>>	filename) : void : throws ImportException, ParseException	new Content item (Application, Ringtone, or Wallpaper) to the Product catalog for each line. Throws ImportException or ParseException for file handling issues or formatting problems with the CSV. *Restricted Interface – must supply a GUID token to import content [currently mocked; all strings accepted]*

SearchEngine

Used to execute content item queries from the Product catalog. Parses CSV file with specified search criteria on each line.

Methods

Method Name	Signature	Description
executeQuery <<static>>	(String : queryLine) : void : throws ParseException	Imports a CSV string, parses the string, and constructs a ContentSearch object. Passes the ContentSearch item to the ProductAPI, which runs the actual query. Throws a ParseException if there is a problem parsing the CSV query string.
executeQueryFilename <<static>>	(String : filename) : void : throws QueryEngineException, ImportException, ParseException	Imports a CSV file and parses each line. Delegates to executeQuery() to run each query. Throws QueryEngineException, ImportException or ParseException for file handling issues or formatting problems with the CSV.

IProductAPI <<interface>>

Interface class defining the public methods of the implementing ProductAPI. Allows customers, administrators, and application developers to search for content items. Allowed authenticated and authorized administrators to add new content to the product catalog by means of a GUID string token passed to restricted methods.

Methods

Method Name	Signature	Description
validateAccessToken	(String : guid) : Boolean	Given the supplied access token, determine if it is valid for accessing the methods of restricted interfaces (e.g., adding content to the product catalog). *Note: currently this method is mocked because the implementation of the Authentication / Authorization service is delayed to a future sprint; this method will return true for any string passed.*
getCountryByCode	(String : code) : Country	Given a 2-character country code, search the product catalog and return any Country that matches that code.

getDeviceByID	(String : deviceID) : Device	Given a device ID string, search the product catalog and return any Device that matches that code.
importCountries	(String : guid, List<Country> countries) : void	Adds new Countries to the product catalog. *Restricted Interface: requires a GUID token to proceed.*
importDevices	(String : guid, List<Device> devices) : void	Adds new Devices to the product catalog. *Restricted Interface: requires a GUID token to proceed.*
importContent	(String : guid, List<Content> contentItems) : void	Adds new Content items to the product catalog (can be Application, Ringtone, or Wallpaper items). *Restricted Interface: requires a GUID token to proceed.*
searchContent	(ContentSearch : search) : List<Content>	Given a ContentSearch object, find all content items that match ANY of the criteria in the ContentSearch object (note this implicitly uses "OR" logic for matching conditions) and returns a list of all those matching items.
getAllApplications	() : List<Application>	Returns all the matching Application content items in the product catalog.
getAllRingtones	() : List<Ringtone>	Returns all the matching Ringtone items in the product catalog.
getAllWallpapers	() : List<Wallpaper>	Returns all the matching Wallpaper items in the product catalog.
getAllContent	() : List<Content>	Returns all the Content items in the product catalog, including all Applications, Ringtones, and Wallpapers.
getCountries	() : Set<Country>	Returns all the Countries in the product catalog.
getDevices	() : Set<Device>	Returns all the Devices in the product catalog.
getNumberContentItems	() : Integer	Returns the total number of content items in the entire product catalog.

ProductAPI

Concrete class implementing the IProductAPI interface. Stores all the content items, countries, devices in-memory. Singleton instance.

Methods

Method Name	Signature	Description
getInstance <<static>>	() : IProductAPI	Static getter. ProductAPI is a singleton, so the only way to get an instance of one is to call this method.

Properties

Property Name	Type	Description
countries	Set<Country>	Holds all the Countries in the entire product catalog in-memory.
devices	Set<Device>	Holds all the Devices in the entire product catalog in-memory.
contentItems	Set<Content>	Holds all the Content items in the entire product catalog in-memory.
instance <<private>>	IProductAPI	Reference to the singleton IProductAPI instance.

Content, Country, Manufacturer, and Device Instance Management

Due to the in-memory nature of this implementation, to optimize memory usage there should only be one instance for each unique Country, Manufacturer, Device, and Content item (where "Content" item is any concrete instance of an Application, Ringtone, or Wallpaper). This follows the Flyweight design pattern (see http://en.wikipedia.org/wiki/Flyweight_pattern).

Implementation Details

The Product API is implemented as a Singleton, as returned by the static getInstance() method. When loading Content, Devices, and Manufacturers into the Product Catalog, it is essential to load the data in the following order:

1. Manufacturers
2. Devices
3. Content

Since Devices contain information on Manufacturers, the Manufacturer data must be loaded first. Likewise, since Content contains information on the Devices that the Content supports, the associated Devices must be loaded into the Product Catalog prior to the Content. Administrators may create Manufacturers, Devices, and then Content through the Product API's public "import*" methods, each of which take a CSV file containing the appropriate data.

When Manufacturers, Devices, and Content are loaded into the Product Catalog from the "import*" methods, new objects of those types are created and maintained in-memory in the Product Catalog (in the "contentItems" property).

To simplify searches of content, the design calls for the creation of ContentSearch objects. When a client of the Product API initiates a new content search via the searchContent() method, the client passes a single string as the search criteria. The searchCriteria() method passes control to parseSearchString(), which returns a ContentSearch object back to searchContent(). Then searchContent() performs the content search over all the contentItems in the Product Catalog based on the criteria in the ContentSearch object, and finally returns a list of all matching Content items.

Changes from Original Design & Requirements

- Allows partial language code searches to match languages that have the criteria (for example, a content search for languages with code “fr” will match content items that have “fr_ca”, French Canadian, as their language code).
- To further emphasize the differences in content item types:
  - Wallpaper has added two properties:
    - pixelWidth
    - pixelHeight
  - Ringtone has added property:
    - durationInSeconds
- To emphasize the separation of concerns, Content items handle the validation of their own types.
- Since Content items are added to collections in the product catalog, to maintain that the ProductAPI only contains a single copy of each item (Device, Country, Application, Ringtone, Wallpaper), each of these types overrides “equals” and “hashCode” to ensure that the Set collections don’t contain duplicates
- This implementation makes use of two Apache Commons JARs to facilitate more easily overriding the “equals” and “hashCode” methods in the content item classes, and also to simplify content searches (the ProductAPI’s searchContent() method makes use of CollectionUtils to compare the intersection of sets).
- Content items (Application, Wallpaper, and Ringtone) and Device and Country classes override the toString() method to simplify debugging, and to more easily observe all the properties of those objects when querying

Testing

The TestDriver class will exercise both importing content into the product catalog via public methods on the ProductAPI as well as searching for specific content items. A static main() method will interactively prompt the user for Countries, Devices, and Content CSV files and call the public import methods on the ProductAPI to load them into the product catalog. Once all required content is loaded, the user will be prompted to supply a CSV query file that will query the API for matching content and output the results to standard out.

In addition to the sample devices.csv, products.csv, and queries.csv that were provided, several new devices, products, and queries were added to further test the functionality and correctness of the implementation.

Risks

As risks are uncovered in the development process they will be documented here.

Instructions for Compiling and Running Application

Because there are two external JAR dependencies, the way to compile and run the application is slightly different from the original specifications.

To compile the code, run the following:

```
javac -cp ".:commons-collections4-4.0-alpha1.jar:commons-lang3-
3.1.jar" cscie97/asn2/ecommerce/product/*.java
cscie97/asn2/ecommerce/product/exception/*.java
cscie97/asn2/test/*.java
```

To run the application, run the following:

```
java -cp ".:commons-collections4-4.0-alpha1.jar:commons-lang3-
3.1.jar" cscie97.asn2.test.TestDriver countries.csv devices.csv
products.csv queries.csv
```