Release 0.2

Description

The second pilot release of DuraCloud is focused on providing access to services which can be run over content, as well as improvements to the storage foundation provided by the first release.

Services available as of release 0.2:

- J2K service - serves J2K images, provides a J2K image viewer
- Image Conversion service - converts image files from one format to another
- Replication service - replicates content stored in one provider to another upon content upload
- Web Application Utility service - infrastructure service required by J2K service (allows for deployment of web applications)
- ImageMagick service - infrastructure service required by Image Conversion service (provides access to ImageMagick utilities)

Service functions available as of release 0.2:

- Services may be deployed with configuration
- Available and deployed services may be listed
- Deployed service configuration may be viewed and updated
- Deployed service properties may be viewed
- Deployed services may be undeployed and redeployed

New storage functions available as of release 0.2:

- Space content may be listed in chunks with an optional prefix filter
- Space and content metadata may be edited via the UI
- Space and content metadata tags may be added/removed via the UI

For more details about specific changes in release 0.2, see the JIRA issue tracker. Note that while most items included in the release are listed in the tracker, we migrated to using JIRA while working on release 0.2, so issues completed prior to the migration are not included.

Deliverables

1. Duradmin instance URL
    - NYPL
        - http://nypl.duracloud.org:8080/duradmin
        - http://nypl.duracloud.org:8080/durastore
    - BHL
        - http://bhl.duracloud.org:8080/duradmin
        - http://bhl.duracloud.org:8080/durastore
    - WGBH
        - http://wgbh.duracloud.org:8080/duradmin
        - http://wgbh.duracloud.org:8080/durastore
2. REST API documentation
3. Store Client - Includes storeclient.jar and all dependent jars as well as Javadocs
4. Simple walk-through instructions for DurAdmin UI
5. JIRA Improvement and Bug tracker
    - Please add issues and bugs here as you find them

DuraCloud REST API (v0.2)

DuraCloud REST API methods:

- DuraStore
    - Get Stores
    - Get Spaces
    - Get Space
    - Get Space Metadata
    - Create Space
    - Set Space Metadata
    - Delete Space
    - Get Content
    - Get Content Metadata

    - Store Content
    - Set Content Metadata
    - Delete Content
- DuraService
    - Get Services
    - Get Service
    - Get Deployed Service
    - Get Deployed Service Properties
    - Deploy Service
    - Update Service Configuration
    - UnDeploy Service

DuraStore

Purpose: DuraStore is the application through which DuraCloud manages storage. The DuraStore REST API provides access to storage by mediating the underlying storage provider APIs to allow access to multiple cloud storage options through a single API.

Store REST Methods

Get Stores

- Purpose: Provides a listing of available storage providers accounts (without credentials)
- Request: GET http://host:port/durastore/stores
- Parameters: None
- Response Code: 200 (on success)
- Response Body: XML similar to:

```
<storageProviderAccounts>
  <storageAcct isPrimary='true'>
    <id>1</id>
    <storageProviderType>AMAZON_S3</storageProviderType>
  </storageAcct>
  <storageAcct isPrimary="false">
    <id>2</id>
    <storageProviderType>RACKSPACE</storageProviderType>
  </storageAcct>
</storageProviderAccounts>
```

Space REST Methods

Get Spaces

- Purpose: Provides a listing of all of the spaces that a customer has created
- Request: GET http://host:port/durastore/spaces ? (storeID)
- Response Code: 200 (on success)
- Response Body: XML similar to:

```
<spaces>
  <space id="space1" />
  <space id="space2" />
</spaces>
```

Get Space

- Purpose: Provides a listing of the contents of a space along with space metadata
- Request: GET http://host:port/durastore/spaceID ? (storeID) (prefix) (maxResults) (marker)
    - storeID (optional) - ID of the content storage provider to query (default is primary store)
    - prefix (optional) - Only retrieve content ids with this prefix (default is all content ids)
    - maxResults (optional) - The maximum number of content IDs to return in the list (default is 1000)
    - marker (optional) - The content ID marking the last item in the previous set (default is the first set of ids)
- Response Code: 200 (on success)
- Response Body: XML similar to:

```
<space id="space1">
  <item>Image 1</item>
  <item>Image 2</item>
</space>
```

- Response Headers: All available space metadata, example:

```
  x-dura-meta-space-count=65
  x-dura-meta-space-access=OPEN
  x-dura-meta-space-created=Mon, 01 Jan 2000 08:00:00 EST
  x-dura-meta-custom-metadata=Custom Metadata Value
```

Get Space Metadata

- Purpose: Provides all space metadata
- Request: HEAD http://host:port/durastore/spaceID ? (storeID)
- Response Code: 200 (on success)
- Response Headers: Same as for Get space (above)

Create Space

- Purpose: Creates a new space
- Request: PUT http://host:port/durastore/spaceID ? (storeID)
- Request Headers: Metadata about the space, example:

```
  x-dura-meta-space-access=OPEN
  x-dura-meta-custom-metadata=Custom Metadata Value
```

- Response Code: 201 (on success)
- Response Headers: Location of the new space (i.e. the URL used to create the space), example:

```
  Location=http://myhost:8080/durastore/space1
```

Set Space Metadata

- Purpose: Updates the metadata associated with a space
- Request: POST http://host:port/durastore/spaceID ? (storeID)
- Request Headers: Same as Create space (above)
- Response Code: 200 (on success)
- Response Body: "Space $spaceID updated successfully" (on success)

Delete Space

- Purpose: Deletes a space
- Request: DELETE http://host:port/durastore/spaceID ? (storeID)
- Response Code: 200 (on success)
- Response Body: "Space $spaceID deleted successfully" (on success)

Content REST Methods

Get Content

- Purpose: Retrieves a piece of content along with its metadata
- Request: GET http://host:port/durastore/spaceID/contentID ? (storeID)
- Response Code: 200 (on success)
- Response Body: The content stream
- Response Headers: All available content metadata, example:

```
Content-Type=text/plain
Content-Length=5732
Content-MD5=3456709234785097473839202
ETag=3456709234785097473839202
x-dura-meta-content-name=Testing Content
x-dura-meta-content-owner=JSmith
```

Get Content Metadata

- Purpose: Retrieves the metadata of a piece of content without the content itself
- Request: HEAD http://host:port/durastore/spaceID/contentID ? (storeID)
- Response Code: 200 (on success)
- Response Headers: Same as Get content (above)

Store Content

- Purpose: Adds a piece of content to the store
- Request: PUT http://host:port/durastore/spaceID/contentID ? (storeID)
- Request Body: Content to be added
- Request Headers: Metadata about the content, example:

```
Content-Type=text/plain
x-dura-meta-content-name=Testing Content
x-dura-meta-content-owner=JSmith
```

- Response Code: 201 (on success)
- Response Headers: Location of the new content (i.e. the URL used to create the content), example:

```
Location=http://myhost:8080/durastore/space1/content1
```

Set Content Metadata

- Purpose: Updates the metadata associated with a piece of content
- Request: POST http://host:port/durastore/spaceID/contentID ? (storeID)
- Request Headers: Same as Store content (above)
- Response Code: 200 (on success)
- Response Body: "Content $contentID updated successfully"

Delete Content

- Purpose: Removes a piece of content from the store
- Request: DELETE http://host:port/durastore/spaceID/contentID ? (storeID)
- Response Code: 200 (on success)
- Response Body: "Content $contentID deleted successfully"

DuraService

Purpose: DuraService is the application through which DuraCloud manages services. The DuraService REST API provides the means by which services available in the DuraCloud service repository are deployed, configured, and undeployed.

Service REST Methods

Get Services

- Purpose: Retrieves a listing of services, along with their configuration options
- Request: GET http://host:port/duraservice/services ? (show)
    - Parameter options for show (optional)
        1. available (default) - Includes only services which have not been deployed but are available for deployment
        2. deployed - Includes only services which have been deployed and started
- Response Code: 200 (on success)
- Response Body: XML list of services (see attached xsd)

Get Service

- Purpose: Retrieves information about a particular service including description, configuration options, and all deployments
- Request: GET http://host:port/duraservice/services/serviceID
- Response Code: 200 (on success)
- Response Body: XML service (see attached xsd)

Get Deployed Service

- Purpose: Retrieves information about a deployed service including description, configuration options, and a single deployment indicating the configuration options in use
- Request: GET http://host:port/duraservice/services/serviceID/deploymentID
- Response Code: 200 (on success)
- Response Body: XML service (see attached xsd)

Get Deployed Service Properties

- Purpose: Retrieves the runtime properties of a deployed service
- Request: GET http://host:port/duraservice/services/serviceID/deploymentID/properties
- Response Code: 200 (on success)
- Response Body: XML service (simple xml Map serialization)

Deploy Service

- Purpose: Deploys and starts an available service
- Request: PUT http://host:port/duraservice/services/serviceID ? (serviceHost)
    - Parameter value for serviceHost (optional) should indicate the services host on which the service should be deployed. Default is the primary customer host.
- Request Body: XML user configuration indicating the config selections for the service (see user config portion of service config xsd)
- Response Code: 201 (on success)
- Response Header: Location header indicates the URL at which information about the deployed service can be retrieved (the URL for a get deployed service call) which includes the deploymentID

Update Service Configuration

- Purpose: Updates the configuration of a deployed service
- Request: POST http://host:port/duraservice/services/serviceID/deploymentID
- Request Body: Updated XML user configuration indicating the config selections for the service (see user config portion of service config xsd)
- Response Code: 200 (on success)

UnDeploy Service

- Purpose: Stops and Undeploys a deployed service
- Request: DELETE http://host:port/duraservice/services/serviceID/deploymentID
- Response Code: 200 (on success)

DurAdmin (v0.2)

Introduction to DurAdmin

The DuraCloud Administrator (DurAdmin) is a web interface designed to provide simple interaction with DuraCloud. DurAdmin sits on top of two DuraCloud applications, DuraStore and DuraService, which handle the content storage and the service deployment portions of DuraCloud, respectively. In this second point release, DurAdmin will be interacting with DuraStore (allowing you to view, update, and delete Spaces and Content) as well as with DuraService (allowing you to deploy several different Services).

Definitions

- **Space** - A top-level container in which content files are stored. Spaces are DuraCloud's single level of file organization. Spaces in DuraCloud are similar in nature to Buckets in Amazon S3 and Containers in the Rackspace Cloud.
- **Content** - A file stored in DuraCloud. Each content item in DuraCloud is stored in a space. The name of a content item can indicate a hierarchical structure (e.g. folder1/folder2/myfile.txt) but DuraCloud treats it simply as a file within a space.
- **Service** - An application or action that, when deployed, will allow for some operation to take place on stored content. Some services are dependent on other services to run properly.

Using the DurAdmin Interface

To use DurAdmin, start by opening a web browser and entering the address provided for your pilot organization. (This should be similar to http://organization.duracloud.org:8080/duradmin/). You will see a welcome page with a small amount of information.

Spaces

Clicking on the ***Spaces*** tab near the top left will bring you to the "Spaces" page. This page lists the spaces that have been created in your account. Hovering over each space will query for the metadata of the space and populate that information on the page. From this page you can:

- Create a new space
    - To create a space use the ***Add Space*** link in the upper left corner of the spaces page. This will give you the opportunity to enter the ***Space Id*** and to select the ***Space Access***. A space can be either OPEN or CLOSED; an OPEN space allows for direct anonymous access to the content in a space through the underlying storage provider; a CLOSED space allows only the space owner to access the content.
- View an existing space
    - View a space by clicking on the space name in the list.
- Add content to an existing space
    - Adding content can be accomplished either on the spaces page or in the space details page. To add content from the spaces page, click on the ***Add Content Item*** link in the space hover menu.
- Delete a space
    - Delete a space by clicking on the ***Remove*** link in the hover menu. Be warned, this is a recursive delete, all of the content items contained in the space will be removed as part of this activity. *Deleting a space CANNOT be undone.*

From the "Spaces" page you can navigate to any one space by clicking on the space name in the list. This will take you to the space details page that will provide you with a listing of all of the content items stored in the space. As on the spaces page, hovering over each content item will retrieve its metadata. From this page you can:

- Space Functions
    - Change the space access property
        - Change the access property of a space by selecting either the ***Open Space*** link (available if the space is currently closed) or the ***Close Space*** link (available if the space is currently open) in the upper right corner.
    - Delete the space
        - Delete the space by clicking on the ***Remove*** link in the top right corner. Please note that this is a recursive delete so all of the content items contained within this space will be removed as part of this activity. *Deleting a space CANNOT be undone.*
    - Add/delete space metadata
        - From the space details bar on the left side of the screen, you can add metadata about the space by clicking on the **+** icon next to the word Metadata.
        - To delete metadata that has been added, simply hover over the metadata and then click the ***x*** icon.
    - Add/delete space tags
        - From the space details bar on the left side of the screen, you can also add tags about the space by clicking on the **+** icon next to the word Tags. To add more than one tag at a time, enter each word separated by the "|" symbol (pipe character).
        - To delete tags that have been added, simply hover over the tag that you wish to delete and then click the ***x*** icon.
- Content Functions
    - Add a content item to the space
        - Add a content item by selecting the ***Add Content Item*** link in the upper left corner. This allows you to select a file and upload it into your space. By default the name of the file will be used as the file's ID, but you can override that value by filling in the ***Content ID*** field. By default the system will attempt to discover the correct MIME type for your file, but you can ensure the correct value by entering it in the ***MIME Type*** field.
    - Delete a content item
        - Delete a content item by selecting the ***Remove*** link in the hover menu. *Deleting a content item CANNOT be undone.*
    - View/edit a content item
        - View more details about a content item by selecting the name of the item or clicking ***Details*** in the hover menu.
        - Selecting the name of a content item will take you to the content details page that provides a detailed view of that item. All of the item's metadata is listed, some of which can be updated. From this page you can:

        - Add/delete content metadata
            - From the content details bar on the left side of the screen, you can add metadata about the content item by clicking on the **+** icon next to the word Metadata.
            - To delete metadata that has been added, simply hover over the metadata and then click the ***x*** icon.
        - Add/delete content tags
            - From the content details bar on the left side of the screen, you can also add tags about the content item by clicking the **+** icon next to the word Tags. To add more than one tag at a time, enter each word separated by the "|" symbol (pipe character).
            - To delete tags that have been added, simply hover over the tag that you wish to delete and then click the ***x*** icon.
        - Edit content MIME Type
            - Again, from the content details bar on the left side of the screen, you can edit the MIME type associated with the content item by including it in the text field and clicking ***Update***. You can update the MIME type as many times as you wish.
        - Add a content item
            - Add a new content item by selecting the ***Add Content Item*** link in the upper-right menu bar. See above for more details.
        - View a content item
            - When the J2K Service has been deployed, image files (files with a MIME type beginning with image/) will be viewable via the J2K viewer. A thumbnail of the image will appear in the box to the right of the content metadata. Clicking on either the thumbnail image or the ***View*** link in the upper-right menu bar will open the viewer. Note that images do not need to be in the J2K format to be viewed, but it will require more time for the thumbnail view and the viewer to display the image. Also note that the larger the file, the longer it will take for the image to render in both the thumbnail view and the viewer.
        - Download a content item
            - Download a file by selecting the ***Download*** link in the upper-right menu bar.
        - Delete a content item
            - Delete a content item by selecting the ***Remove*** link in the upper-right menu bar. *Deleting a content item CANNOT be undone.*

Services

Clicking on the ***Services*** tab near the top left will bring you to the "Services" page. By default, this page lists the services that have been deployed in your account (if no services have been deployed the page will be blank). From this page you can:

- View/reconfigure/undeploy deployed services
    - By default, the services that have been deployed as well as the hostname and status of the service are listed on the services page. To view more information about a deployed service, click the ***View*** link on the right side of the page. This will bring up a detailed description box underneath the service.
    - Please note that certain services allow you to reconfigure the deployment. If this is an option, the ***Reconfigure*** link will appear on the right side of the page.
    - To undeploy a service, simply click the ***Undeploy*** button.
- View available services
    - To view available services, click the ***Available*** link in the upper left corner of the services page. From the "Available Services" page you can read a description of the available services and choose to deploy the service by clicking on the ***Deploy New Instance*** link. Please note that for this release, the service can only be deployed to the primary service instance.

Storage Providers

To navigate between the available storage providers, use the dropdown menu in the top right of the screen. For this release, the default service provider is Amazon S3 and the only other storage provider option is Rackspace.

Available Services

- **Image Conversion Service** - The Image Conversion service provides a simple way to convert image files from one format to another. A space is selected from which image files will be read and converted to the chosen format. The converted image files will be stored in the destination space along with a file which details the results of the conversion process. Note that the ImageMagick service must be deployed prior to using the Image Conversion service.
- **ImageMagick Service** - The ImageMagick service deploys the ImageMagick application which allows other services to take advantage of its features.
- **J2K Service** - The J2K service deploys an instance of the Adore Djatoka web application which provides for serving and viewing JPEG2000 images.
- **Replication Service** - The Replication service provides a simple mechanism for synchronizing your content between two storage providers. A running replication service will listen for updates which occur in one store and duplicate those activities in another store.
    - For this release the replication service
        - will only start replicating after you turn on the service (nothing that exists prior to the service being deployed will be replicated)
        - will only replicate content upon initial upload
- **Web App Utility Service** - The Web App Utility service coordinates the (de)installation and startup/shutdown of Apache Tomcat instances that are created to run web application services that are deployed externally to the hosting OSGi container.