Source Documents - Android SDK - Content Management

Android SDK Integration Guide - Content Management

Version 1.1.3

ON THIS PAGE

This is Phunware's Android SDK for the Content Management module. Visit http://maas.phunware.com/ for more details and to sign up.

Requirements

- Android SDK 2.2+ (API level 8) or above
- Android Target 4.1.1.4
- Android Support v4 18.0.+

Getting Started

- Download MaaS Content Management and run the included sample app.
- Continue reading below for installation and integration instructions.

Installation

The following libraries are required:

```
MaaSCore.jar
```

MaaS Content Management has a dependency on MaaSCore.jar, which is available here: Core SDK

Overview

The MaaS Content Management SDK allows developers to fetch and manage the various pieces of data in the Content Management Engine, including containers, schemas, structure and content. MaaS Content Management spans across your entire organization, so different applications can potentially share the same content.

Container

Containers hold a single structure. You can create any number of containers in the MaaS portal. You can also associate tags with containers to assist with fetching.

Schema

Schemas are applied to **structure** items and define what fields of data a particular structure item can contain. You can create any number of schemas in the MaaS portal. You can also associate tags with schemas to assist with fetching.

Structure

Structure items are used to build the structure and hierarchy of the data. Each **structure** item that is defined as an object can also optionally be assigned a **schema** that defines what content can be saved to those **structure** items.

Content

The structure of the **content** object relies completely on the layout of structures and schemas.

Integration

The primary methods in MaaS Content Management revolve fetching, creating, updating and deleting content. You can also get structures, containers and schemas.

Getting Content

```
// Get a specific piece of
content for the specified
context, container ID and
content ID.
 // The contents are always
returned as a JSONObject. It's
recommended that you parse the
JSONObject into a model
object.
    JSONObject content =
PwCMEModule.getContent(this,
containerId, contentId);
```

Updating Content

```
// Update content for the
specified content ID,
container ID and structure ID.

 // Any fields that are
omitted will maintain their
previous values.

PwCMEModule.updateContent(this
, contentId, newContent,
structureId);
```

Creating Content

```
// Add content to the
specified container ID,
structure ID and parent
content ID.
 // Ideally, the new content
object has all the fields
specified by the structure and
schema.
 // If not, the required
fields will be created for you
with empty values.

   // The required parent
content ID for new content
needs to be linked up to any
dynamic children of a
structure item.

PwCMEModule.addContent(this,
containerId, structureId,
parentId, data);
    or
    // If no parent content
exists, then the parent
content ID is not required.
    // Otherwise, use the
above method to properly link
child elements.

PwCMEModule.addContent(this,
containerId, structureId,
data);
```

Deleting Content

```
// Delete content for the
specified content ID as well
as all content children.

PwCMEModule.deleteContent(this
, contentId, traverse);

    or
  // Delete all content
children for the specified
content ID.

PwCMEModule.deleteContentAllCh
ildren(context, contentId);
```

Containers

```
// Fetch all containers.
    PwContainers containers =
PwCMEModule.getContainers(this
);

  // Fetch a specific
container item.
    PwContainer container =
PwCMEModule.getContainer(this,
containerId);

  // Get an array of
containers that match an array
of tags.
    PwContainers containers =
PwCMEModule.getContainers(this
, anyTags, allTags);
```

Schemas

```
// Fetch all schemas.
    PwSchemas schemas =
PwCMEModule.getSchemas(this);

  // Fetch a specific schema
item.
    PwSchema schema =
PwCMEModule.getSchema(this,
schemaId);

  // Get an array of schemas
that match an array of tags.
    PwSchemas schemas =
PwCMEModule.getSchemas(this,
anyTags, allTags);
```

Structure

```
// Fetch a structure with the
specified stucture and
container ID.
  // In this example, we want
to traverse into all child
structures but not include
schema.
    withSchema = false;
    PwStructure structure =
PwCMEModule.getStructure(this,
structureId, containerId,
depth, withSchema);

  // Fetch a structure object
containing an array of
structures for the specified
container ID.
  // In this example, we want
to traverse into all child
structures and include schema.
    withSchema = true;
    PwStructures structures =
PwCMEModule.getStructures(this
, containerId, depth,
withSchema);
```