Web Services

Local Shop

Introduction

The Pro version of the Moodle shop provides Web services allowing partnership operations to be run and the sales catalogue to be exposed to third-party services.

Implementing the web services

To enable the web services, you will need to:

- Globally enable Moodle web services
- Enable the “Moodle Shop Definition Access API” service
- Add a user authorised for this service
- Assign the capability local/shop:export to this user through a system role assignment (if the user is not an administrator)
- Create a service “token” for the service and the user.

Web Services catalogue

local_shop_get_shop(shopid)

Synopsis: Get the definition of a shop instance.

- **shopid**: The instance identifier (starts at 1)

RETURN

- A shop descriptor object

General response structure:

```
object {
   id int   //Shop id
   name string   //Shop name
   catalogid string   //Master shop catalog
   description string   //Shop description
   allowtax int   //Do the shop apply VAT tax
```

```
   eulas string   //Shop eulas
   paymodes string   //Enabled paymodes
   defaultpaymode string   //Default paymode
}
```

local_shop_get_catalog(catalogid)

Synopsis: Get the definition of a catalogue instance.

- **catalogid**: The catalogue identifier (starts at 1)

RETURN

- A catalogue descriptor object

General structure

```
object {
   id int   //Catalog id
   name string   //Catalog name
   description string   //Catalog description
   salesconditions string   //Catalog Eulas
   countryrestrictions string   //Countries deserved
   categories list of (
      object {
         id int   //Category id
         name string   //Category name
      }
   )
}
```

local_shop_get_catalogcategory(categoryid)

Synopsis: Get the definition of a catalogue category.

- **categoryid**: The category identifier (starts at 1)

RETURN

- A category descriptor object

General structure

```
object {
   id int   //Category id
   catalogid int   //Catalog id
   name string   //Category name
   description string   //Category description
   visible int   //Is category visible
```

```
}
```

local_shop_get_catalogitem(itemcourse, itemid, q)

Synopsis: Get the definition of a product.

- **itemsource**: The reference field for product identification ('id', 'code', 'shortname')
- **itemid**: The product identifier, according to the chosen source field.
- **q**: The quantity used to compute the unit price of the product for this quantity.

RETURN

- A product descriptor object

General structure

```
object {
   id int   //Item id
   catalogid int   //Catalog id
   code string   //Item code
   shortname string   //Item shortname (for web UI)
   name string   //Item name
   description string   //Item description
   eulas string   //Item eulas
   notes string   //Item notes
   type string   //Item type, plain, set or bundle
   status string   //Item status
   unitcost string   //Unit cost for input quantity
   tax string   //Tax cost
   requireddata string   //Required data from the front customer
   leafleturl string   //Leaflet url
   thumburl string   //Thumb url
   imageurl string   //Image url
}
```

local_shop_get_catalogitems(catalogid, categoryid, status, type, q)

Synopsis: Get a list of products

- **catalogid**: The catalogue identifier, '*' for all catalogues.
- **categoryid**: The category identifier, '*' for all categories.
- **status**: The product status: 'PREVIEW', 'AVAILABLE', 'AVAILABLEINTERNAL', 'SUSPENDED', 'PROVIDING', 'ABANDONNED'.
- **type**: The product type: 'plain', 'set' or 'bundle', or '*' for all types.
- **q**: The quantity used to compute the unit price of the product for this quantity.

RETURN

- A product descriptor object

General structure

```
list of (
   object {
      id int   //Item id
      catalogid int   //Item catalog id
      categoryid int   //Item category id
      code string   //Item code
      shortname string   //Item shortname (for web UI)
      name string   //Item name
      description string   //Item description
      eulas string   //Item eulas
      notes string   //Item notes
      type string   //Item type, plain, set or bundle
      status string   //Item status
      unitcost string   //Unit cost for input quantity
      tax string   //Tax cost
      requireddata string   //Required data from the front customer
      leafleturl string   //Leaflet url
      thumburl string   //Thumb url
      imageurl string   //Image url
   }
)
```

Back to the technical guide summary - Back to the component index

From:
https://docsen.activeprolearn.com/ - **Documentation Moodle ActiveProLearn**

Permanent link:
https://docsen.activeprolearn.com/doku.php?id=local:shop:technique:webservices

Last update: **2026/11/13 21:59**