

# CommunityMashup - Applications

## Basics

In its core the CommunityMashup is an OSGi-based Service Oriented Architecture, that uses a custom [EMF meta model](#) as shared Data Set.

A CommunityMashup instance can be accessed via Web Browser (see below).

However, usually you will build applications to access a mashup. So, in the rest of the page we will describe how the CommunityMashup can be used from a variety of programming languages / application environments.

## Accessing a CommunityMashup via REST API

You can use a CommunityMashup instance in an application via the mashups interfaces.

However, you can also directly access these interfaces from a web browser (or more generically via a REST API). The Data Set contents are published by the REST API in [XML Metadata Interchange \(XMI\)](#) format by default.

See the restdoc section in the CommunityMashup instances at [communitymashup.net](http://ulp.communitymashup.net/restdoc/data_overview.html) for more information: [http://ulp.communitymashup.net/restdoc/data\\_overview.html](http://ulp.communitymashup.net/restdoc/data_overview.html)

## CommunityMashup: Item

The Data Set contains a collection of Items consisting of:

- Information Objects,
- Extensions,
- Classifications, and
- Mashup Tags.

An Item defines REST-compliant operations that can be called by GET, POST, PUT or DELETE. The result of an operation call is either XMI content, a Base64url-encoded serialized Java Object or a String.

## CommunityMashup: Ident

The unique identification property defined for all Items is called an *ident*.

Items can be selected by its identifier by calling the operation *getItemsWithIdent*. `/getItemsWithIdent?ident=<identifier>`  
In addition several operation calls can be defined as a sequence in one query string: `/getItemsWithIdent?ident=<identifier>/getImages/`

The behaviour of these operation calls can be tested by the given examples in the following section.

## CommunityMashup: Information Object

One of the main Items of the person-centric CommunityMashup meta model is the Information Object. An Information Object is an abstract superclass for Persons, Organisations and Content.

## Examples

The following table lists several examples that can be tested against the CommunityMashup Data Set. The Java column shows how these calls are performed in Java. The REST column shows how the same operations are called as stateful resources. The result column describes the type of data returned by the REST operation call.

| Java  | REST   | Result   |
|---|--|--|
| DataSet                                     | /mashup  | The overall CommunityMashup Data Set.          |
| DataSet.getItemsWithIdent("pl")             | /mashup/getItemsWithIdent?ident=pl                 | The Item with ident "pl".                      |
| DataSet.getItemsWithIdent("pl").getImages() | mashup/getItemsWithIdent?ident=pl<br>/getImages    | A list of Images for the Item with ident "pl". |
| DataSet.getPersons()                        | /mashup/getPersons                                 | A list of Persons described by the meta model. |
| DataSet.getPersonWithIdent("mb")            | /mashup/getPersonWithIdent?ident=mb                | The Person with ident "mb".                    |
| DataSet.getPersonsWithLastname("Koch")      | /mashup/getPersonsWithLastname?<br>lastname=Koch   | A list of Persons with lastname "Koch".        |
|   | /mashup/getMetaTag?name=gravatar<br>/getMetaTagged |  |

More operations are described in the operations section of the Data Set documentation.

### Important Remarks:

- In case an Item does not exist, the result is empty. No error status or message is displayed.
- The REST operation names and field names of the query string are **not** case-sensitive, whereas the field values are case-sensitive:

```
/mashup/getitemswithident?ident=pl is correct.  
/mashup/getItemsWithIdent?IDENT=pl is correct.  
/mashup/getItemsWithIdent?ident=PL is not correct!
```

## Accessing a CommunityMashup via a Web Browser - HTML endpoint

The REST API can be accessed via a standard web browser. However, the API usually returns content in XML or JSON - not very human readable.

Additionally, it is possible to define a HTML endpoint and get HTML content in return:

URL: <http://domain/html/mashup/>

In the HTML interface you can use templates and use some modifiers:

URL: <http://domain/html/mashup/getOrganisations?tpl=csv&wrap=false>

TBD

## Developing Mashup Applications

### Interfaces

Access to a mashup instance can be done directly using the mashup interfaces. We currently provide interfaces for

- HTML (over HTTP)
- XML (over HTTP)
- JSON (over HTTP)
- REST

The interfaces are configured in the mashup configuration.

## Frameworks

Using the data model of the mashup it is possible to easily generate so called coding frameworks for accessing a mashup via the XML interface for different programming languages.

We currently provide frameworks for:

- JavaScript (for developing Web applications)
- iOS
- Android
- PHP
- ...

See the corresponding sub-pages for more information and for code examples.