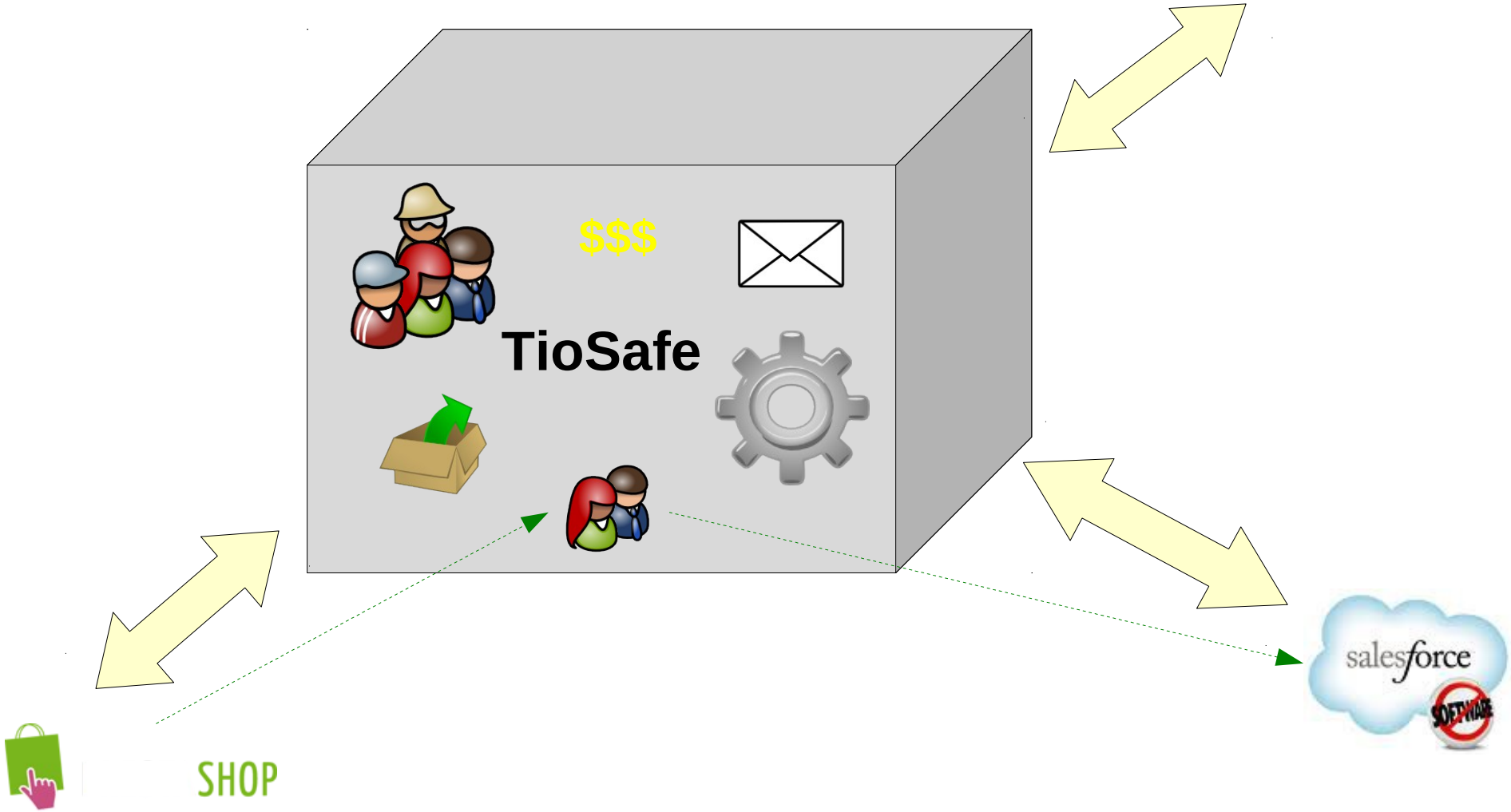
TioSafe Connector Howto

This guide will teach you:

- What is TioSafe
- What is a TioSafe plugin
- How to create a TioSafe connector

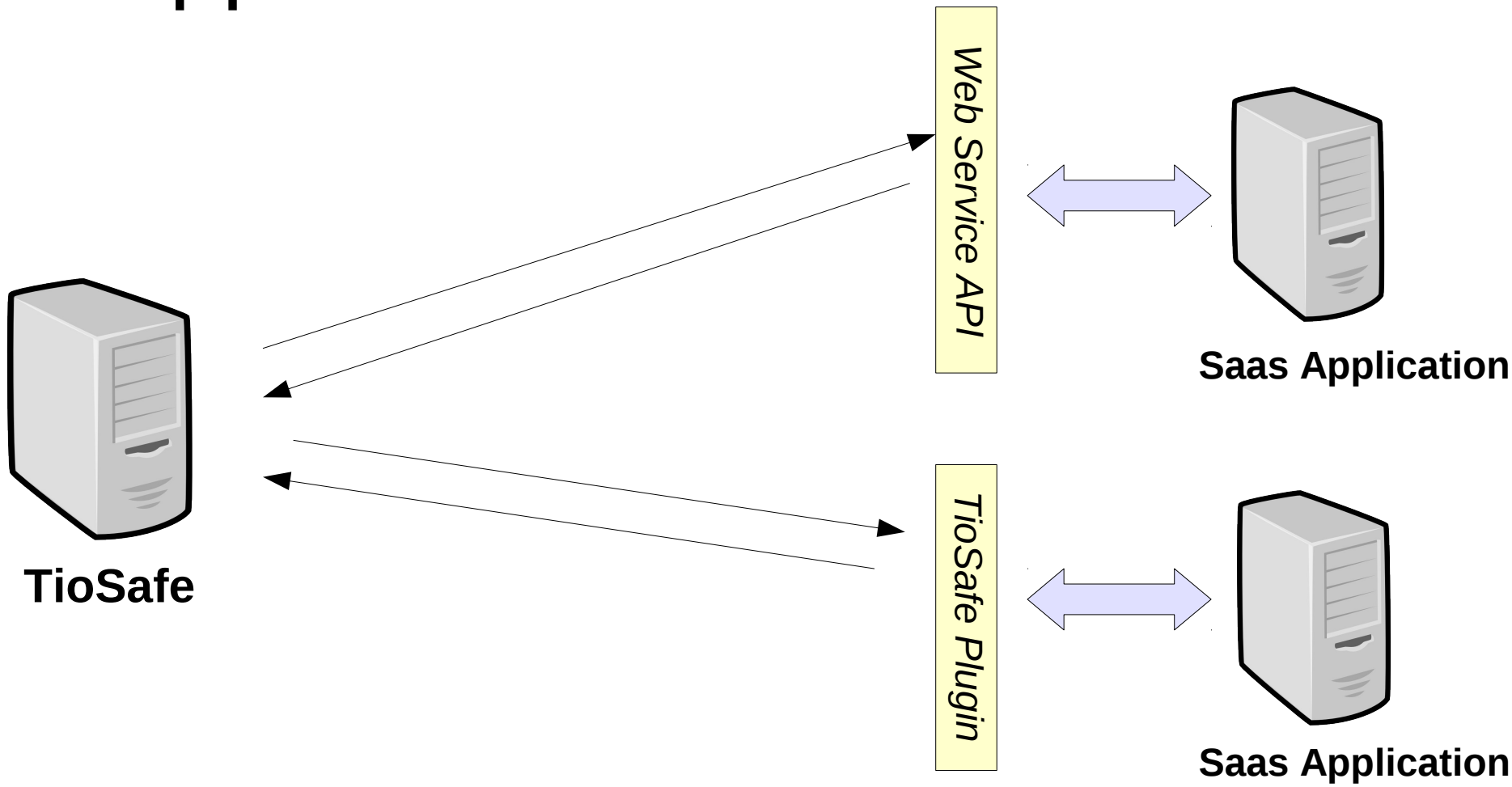
© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

What is TioSafe



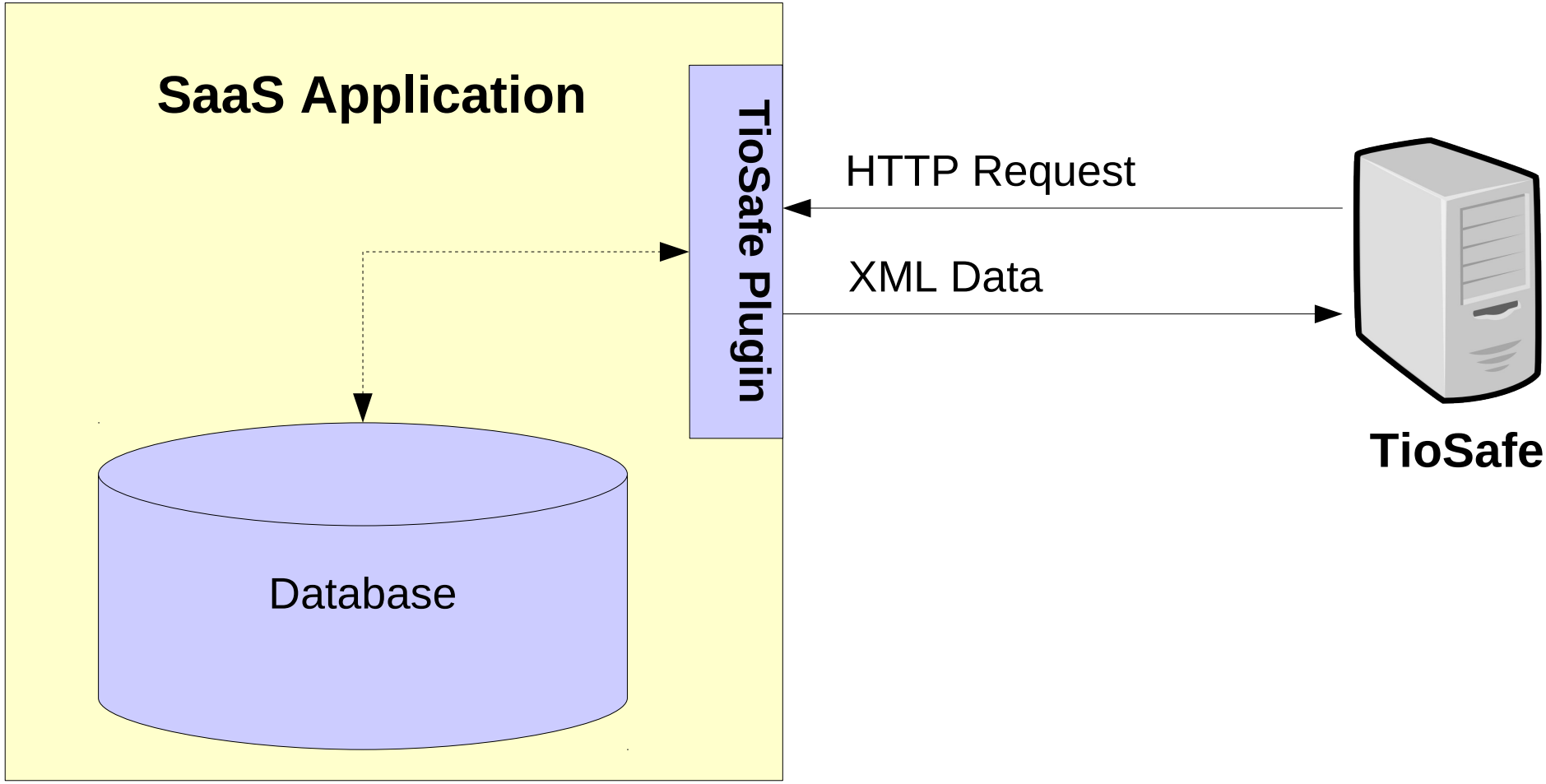
© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

SaaS Application Access



© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

TioSafe Plugin



© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

What a TioSafe plugin must do

- Provides access to data from application based on simple http request :
  - Get data
  - Put data
  - Update data
  - Delete data
- Supports authentification if not provided by the application
- Act as an oauth client regarding the TioSafe server

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Which data must be retrieved

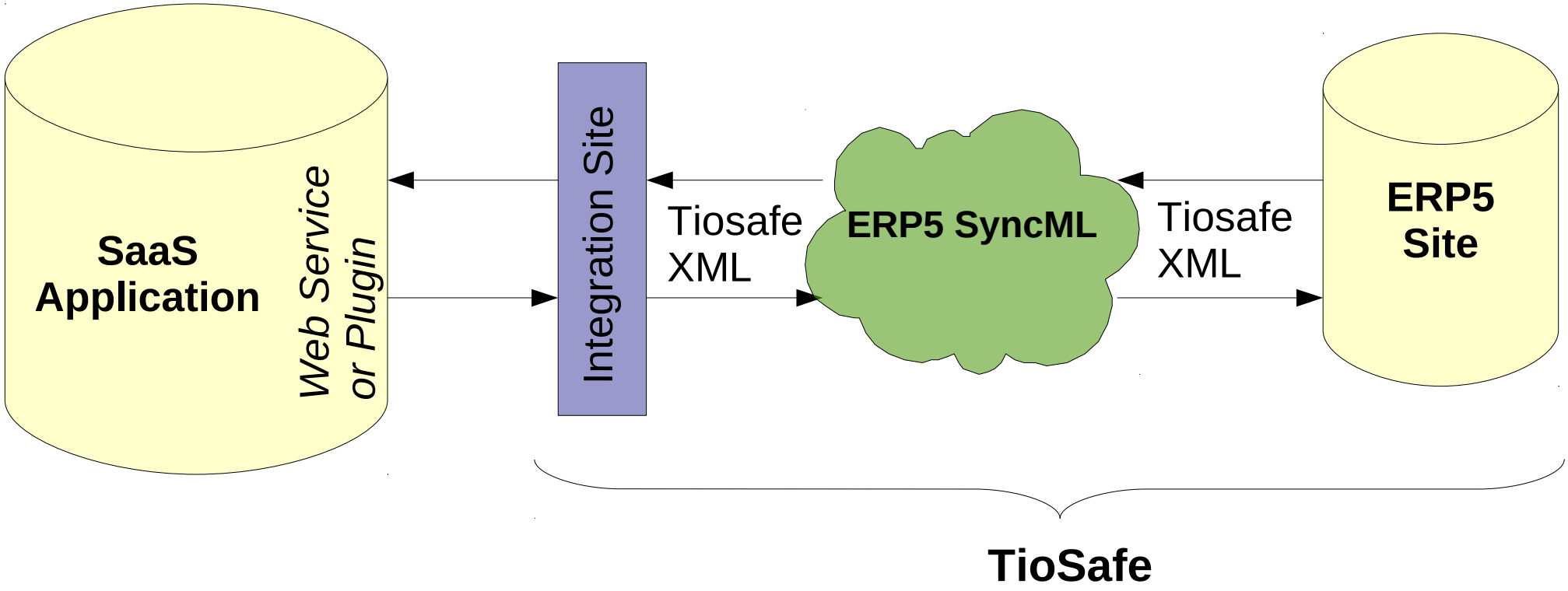
Data returns by plugin or web service calls depends on object type :

- Nodes : Persons, Organisations, Accounts
- Resources : Products, Services, Components
- Transactions : Orders, Events, Accounting transactions, Packing Lists

All properties exchanged are defined in XSD files contains in the ERP5TioSafe products

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

TioSafe Global View



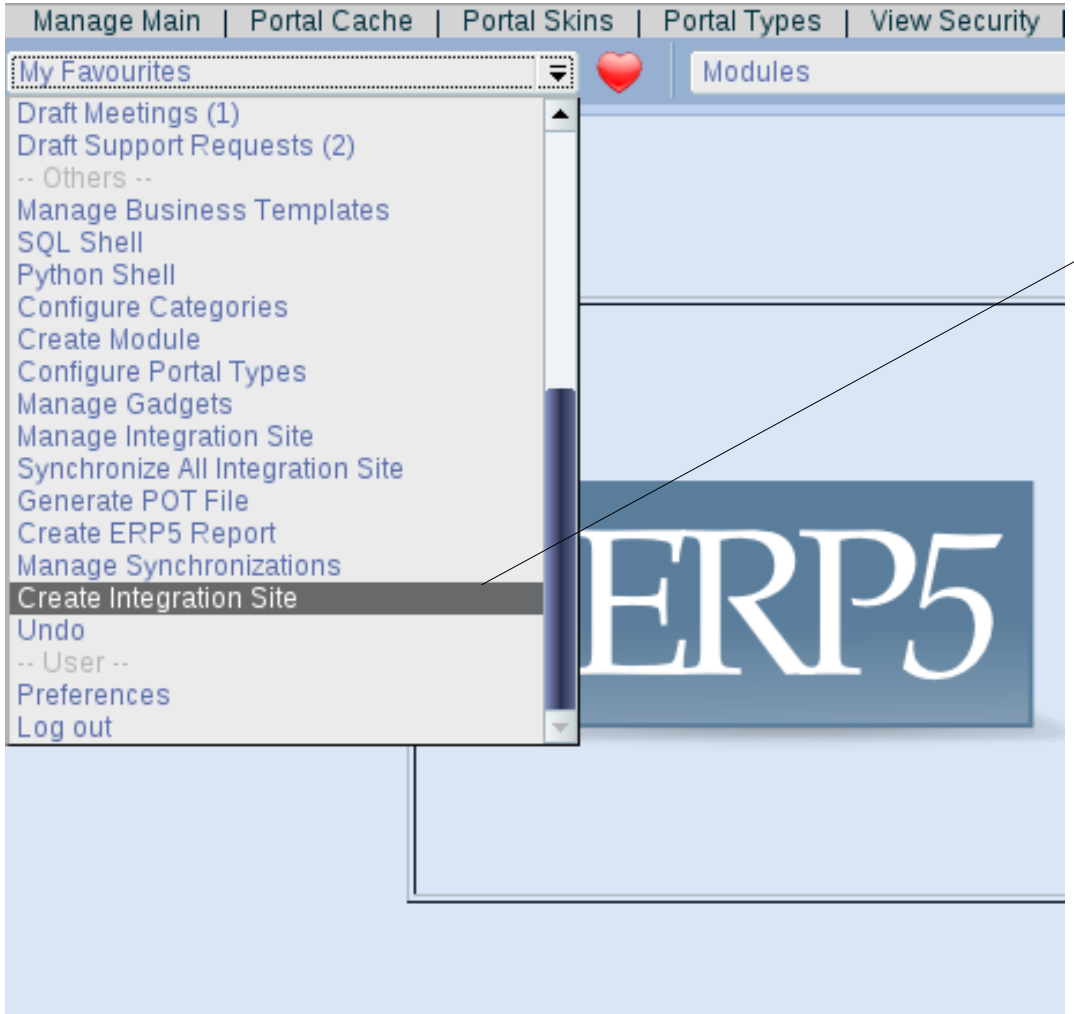
© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Requirements

TioLive instance + ERP5TioSafe Product + erp5_tiosafe_core bt5

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Creating an Integration Site



This will redirect to the form used to create a new integration site

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Integration Site Creation Form

Type of connections used to access the Web Service or Plugin

Lists of modules that will be synchronized

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Integration Site Created

Integration Site created.

View | Synchronization Configuration | Mapping | History | Metadata

Title	Magento
Reference	magento
Language	
Default Sale Trade Condition	default_sale_trade_condition
Default Purchase Trade Condition	default_purchase_trade_condition
Default Organisation	
Default Discount Service	default_discount
Default Delivery Service	default_delivery
Start Date	/ /
Start Date	/ /

Contents: 13 records

ID	Title	Type	Reference
IntegrationSite_createPerson	IntegrationSite_createPerson	Web Service Request	
IntegrationSite_createProduct	IntegrationSite_createProduct	Web Service Request	
IntegrationSite_createSaleOrder	IntegrationSite_createSaleOrder	Web Service Request	
IntegrationSite_deletePerson	IntegrationSite_deletePerson	Web Service Request	
IntegrationSite_deleteProduct	IntegrationSite_deleteProduct	Web Service Request	
IntegrationSite_deleteSaleOrder	IntegrationSite_deleteSaleOrder	Web Service Request	
IntegrationSite_updatePerson	IntegrationSite_updatePerson	Web Service Request	
IntegrationSite_updateProduct	IntegrationSite_updateProduct	Web Service Request	
IntegrationSite_updateSaleOrder	IntegrationSite_updateSaleOrder	Web Service Request	
http_connector	http_connector	Web Service Connector	
person_module	person_module	Web Service Request	
product_module	product_module	Web Service Request	
sale_order_module	sale_order_module	Web Service Request	

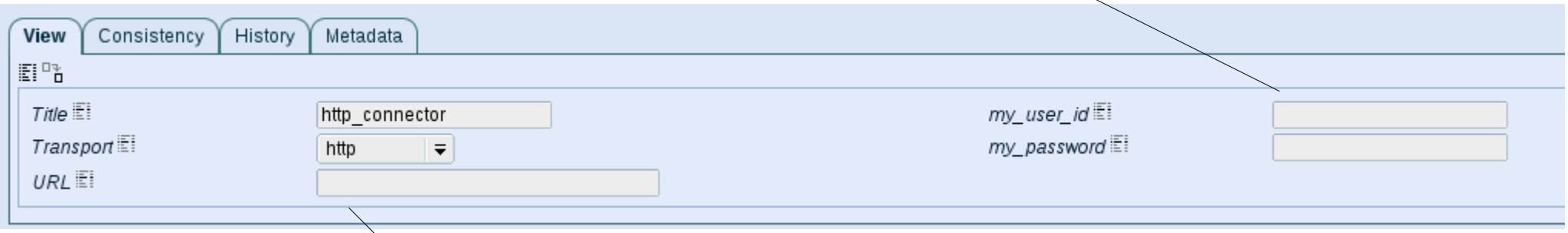
Default configuration, some fields will updated by final users

All required Web Service Requests and Connectors are created automatically

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Web Service Connector

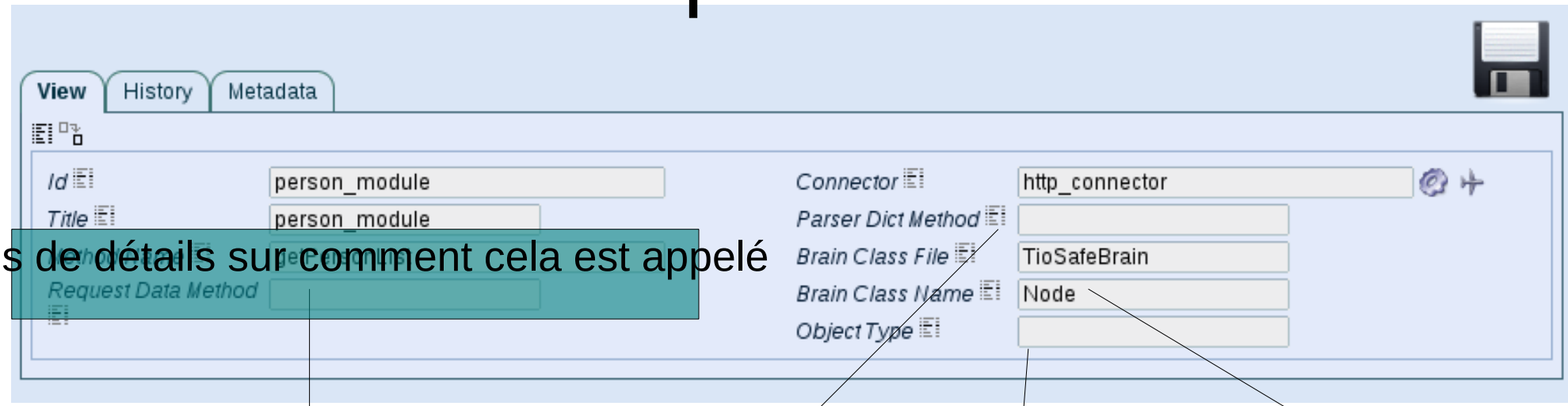
Authentification parameters



URL which provides the service
(must be a complete url that include the plugin path)

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Web Service Request



Plus de détails sur comment cela est appelé

This defined the method called on the application web service

If web service returns xml, this defined the script used to parse it and return a dict of properties

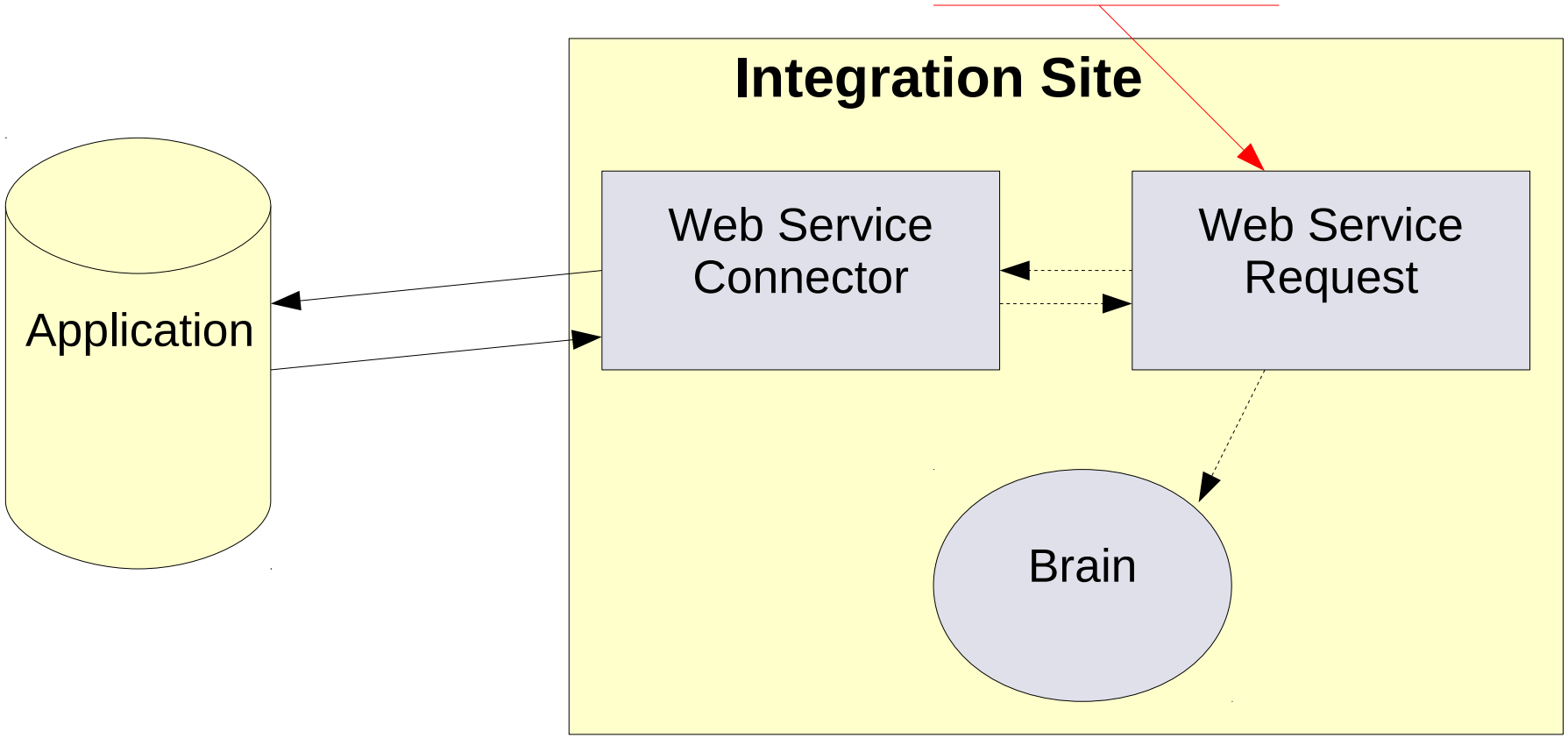
Type of object that will be create in ERP5

Default brain classes should managed most of the cases. If not, you can define here specific classes.

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Web Service Request

url = portal_integration/integration_site/person_module/1



© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

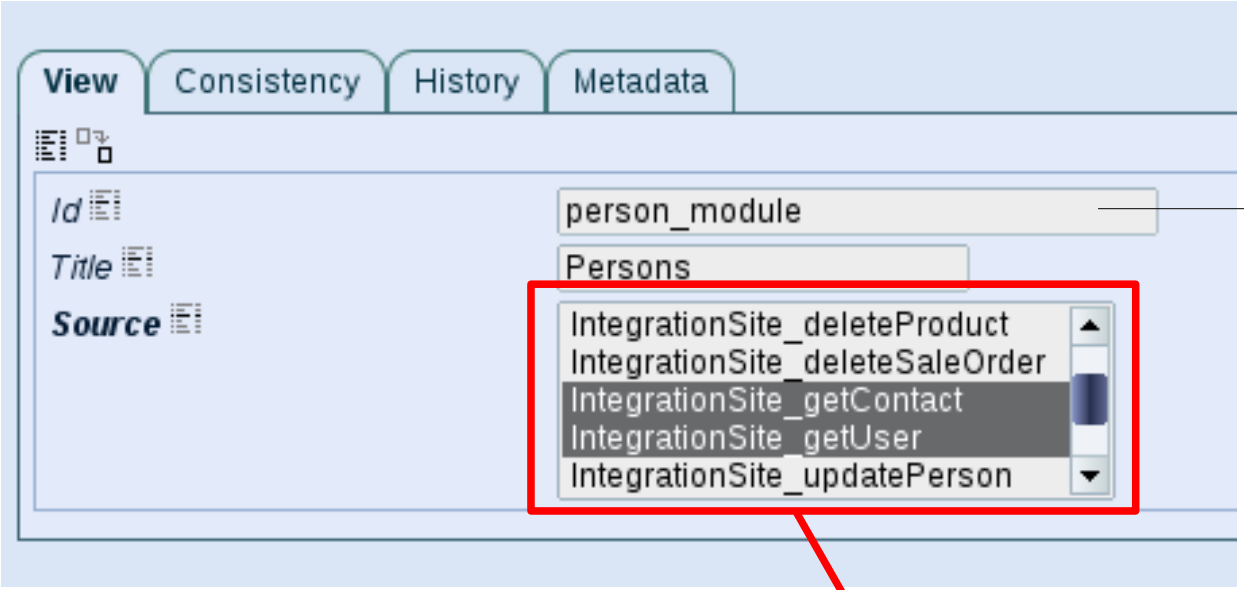
How to define ID of WSR

portal-type : Type of the object in TioSafe

- *portal-type*_module : simulate a TioSafe module, used to retrieve object
- IntegrationSite_create*Portal-Type* : method called to create object in the application
- IntegrationSite_update*Portal-Type* : method called to update object in the application
- IntegrationSite_delete*Portal-Type* : method called to delete object in the application
- IntegrationSite_get*Portal-Type*LastID : method called to get the last ID in the application for Portal-Type

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Web Service Request Group



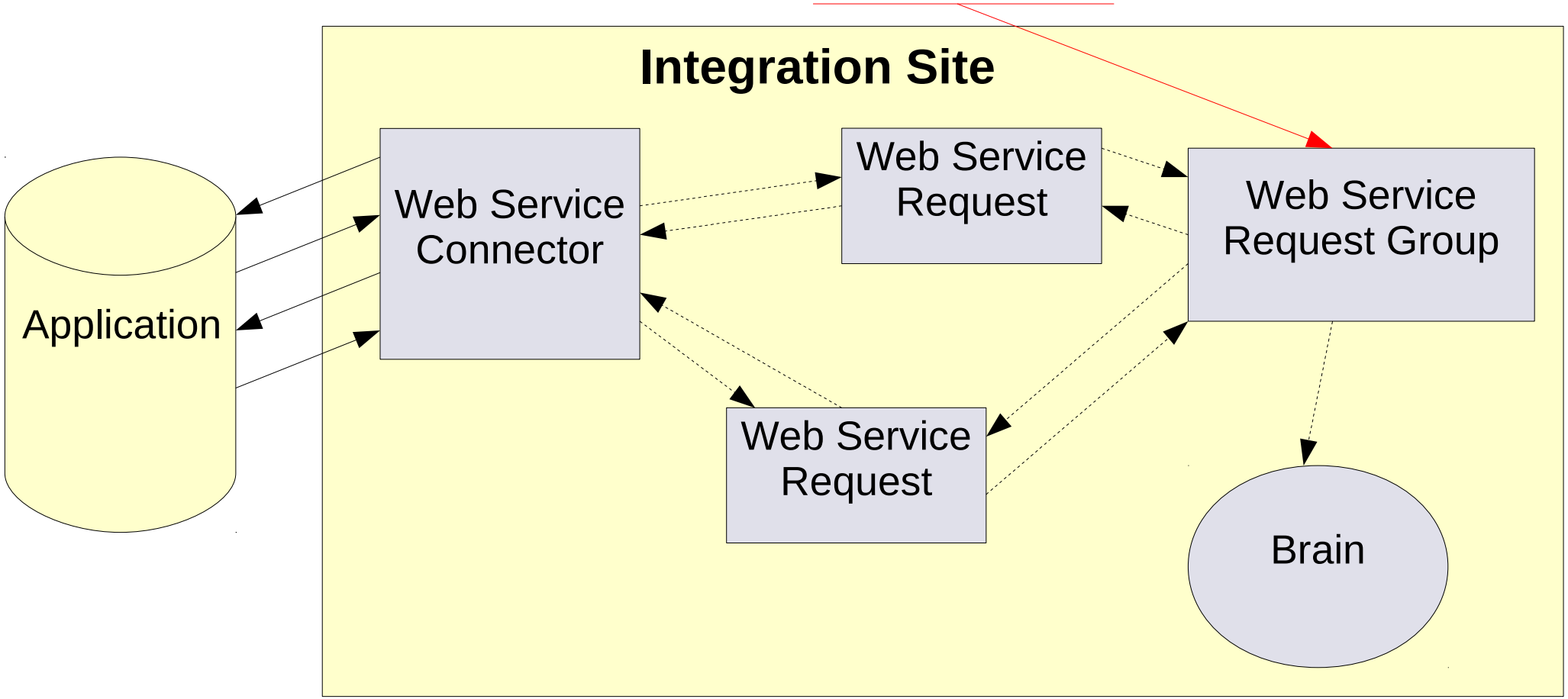
ID of the method called by TioSafe

List of Web Service Requests used

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

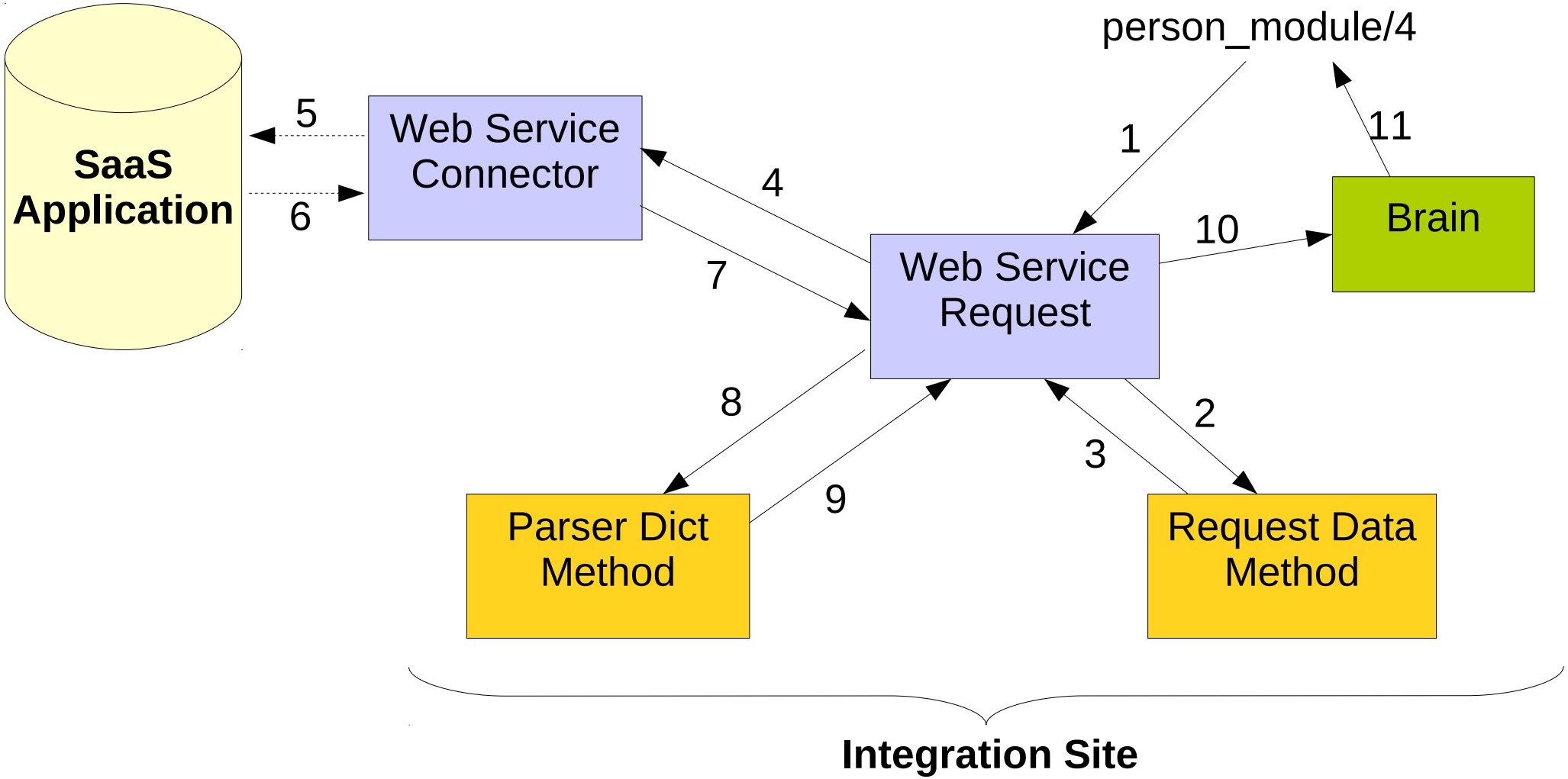
Web Service Request Group

url = portal_integration/integration_site/person_module/1



© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

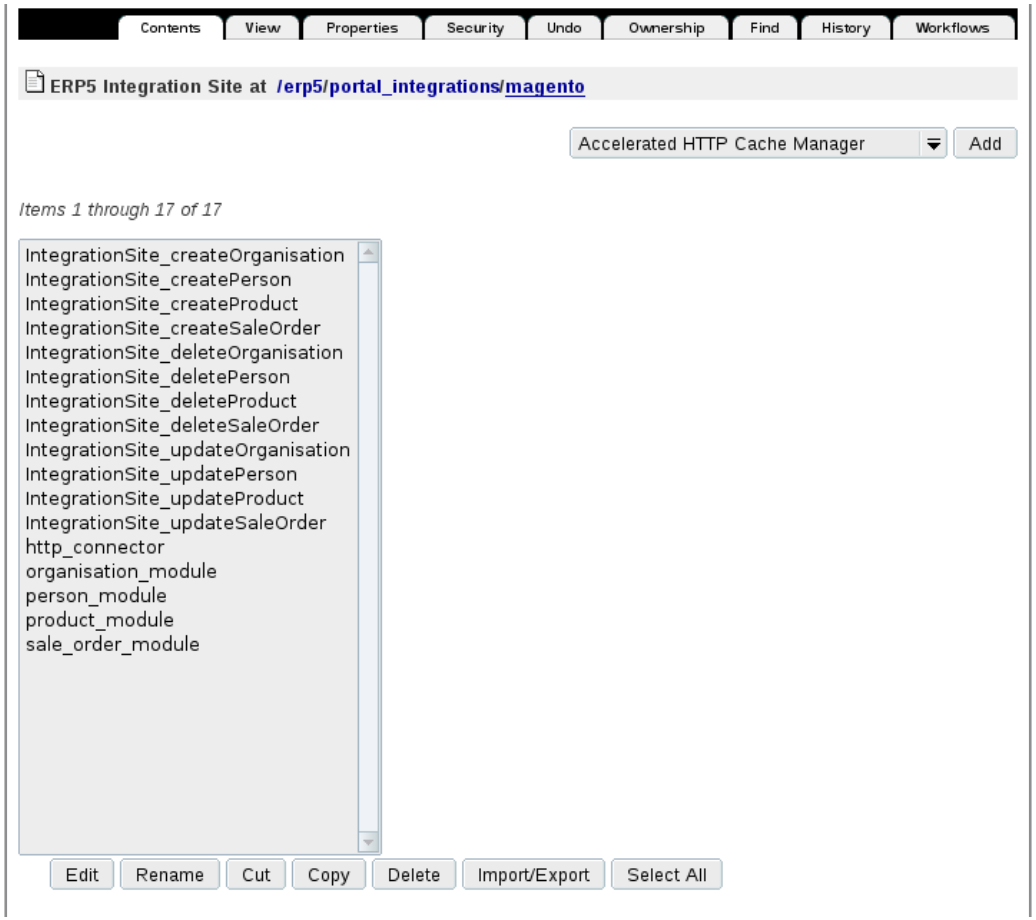
Web Service Request Process



© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

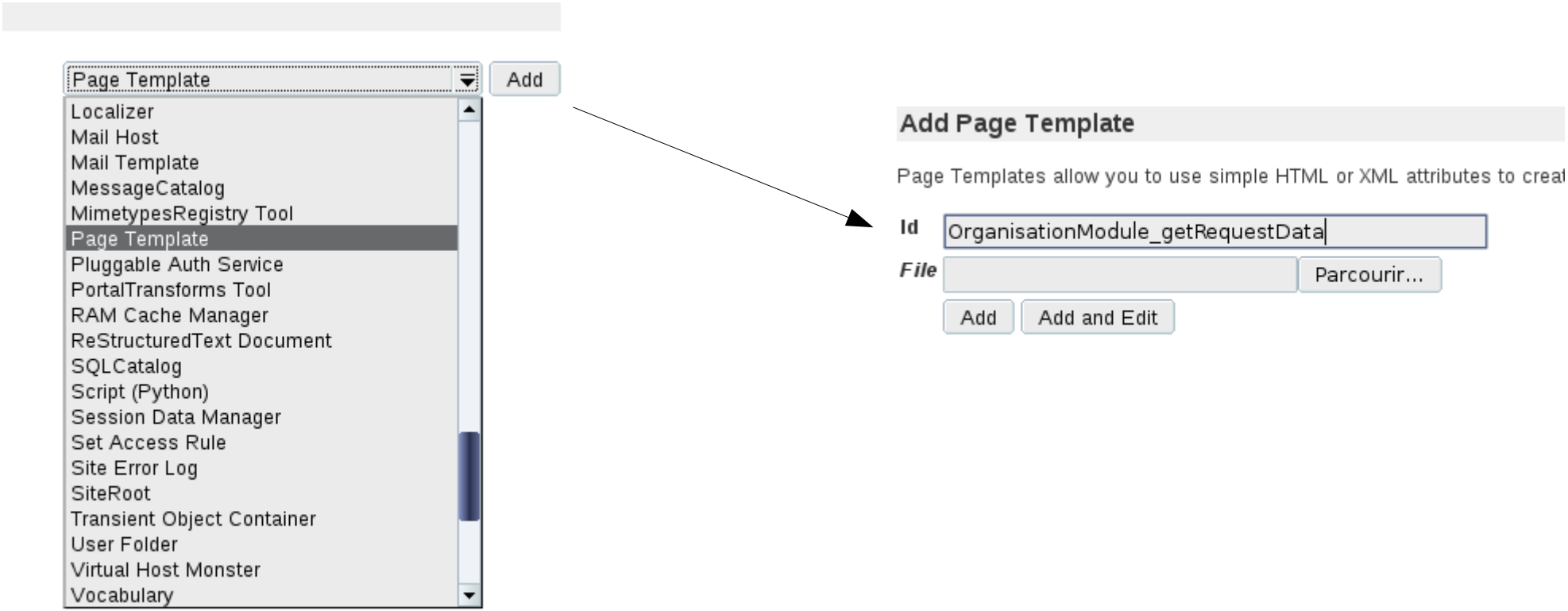
Access to the manager view

http://localhost:8080/erp5/portal_integrations/magento/manage_main



© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Request Data Method Add



© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Request Data Method Edit

Page Template at /erp5/portal_integrations/magento/OrganisationModule_getRequestData

Title

Content-Type text/html

Browse HTML source

Expand macros when editing

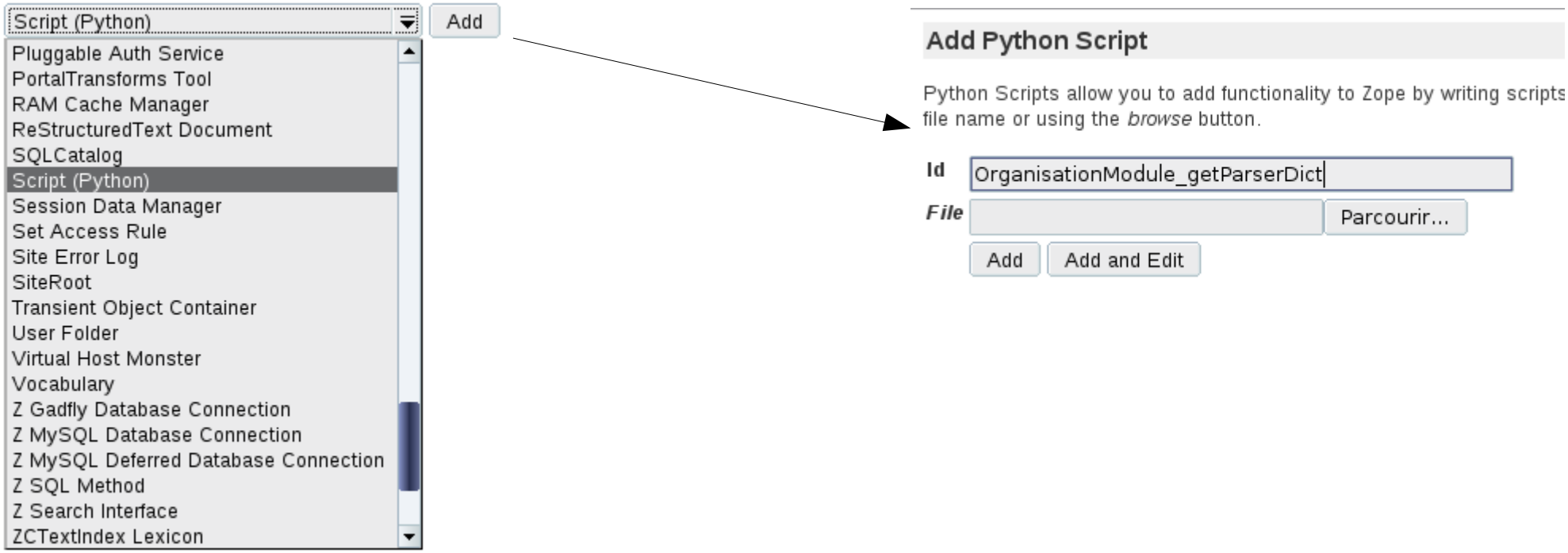
Last Modified 2010-07-01 03:04 PM

```
<ns0:query>
  <ns0:queryString>
    SELECT
      Id, Name, FirstName, LastName, Email, Phone, MobilePhone, Birthdate
    FROM
      Contact
    WHERE
      <tal:block tal:condition="python: options['id'] is not None">Id = '<tal:block tal:content="options/id" />' AND</tal:block> Email != NULL
  </ns0:queryString>
</ns0:query>
```

Example of body of a SOAP Request

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Parser Dict Method Add



© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Parser Dict Method Edit

Edit | Bindings | Test | Proxy

Script (Python) at /erp5/portal_integrations/magento/OrganisationModule_getParserDict

Title

Parameter List

Bound Names context, container, script, traverse_subpath

Last Modified 2010-07-01 15:16

```
# Parse the result
base_tag = ""
parser_dict = {
      base_tag+'object' : ('object', True),
      base_tag+'PATH' : ('path', False),
      base_tag+'title' : ('title', False),
      base_tag+'firstname' : ('firstname', False),
      base_tag+'lastname' : ('lastname', False),
      base_tag+'email' : ('email', False),
      base_tag+'birthday' : ('birthday', False),
      base_tag+'category' : ('category', False),
      base_tag+'gid' : ('gid', False),
}
return parser_dict
```

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Parser Dict Method

```
<object>
    <name>Bag</name>
    <product-reference>bg-12314</reference>
</object>
```

XML Returned by the plugin

```
parser_dict = {
    'object' : ('object', True),
    'product-reference' : ('reference', False),
    'name' : ('title', False)}
```

IntegrationSite_getPortalTypeParserDict

indicate that this tag is the root of a new element

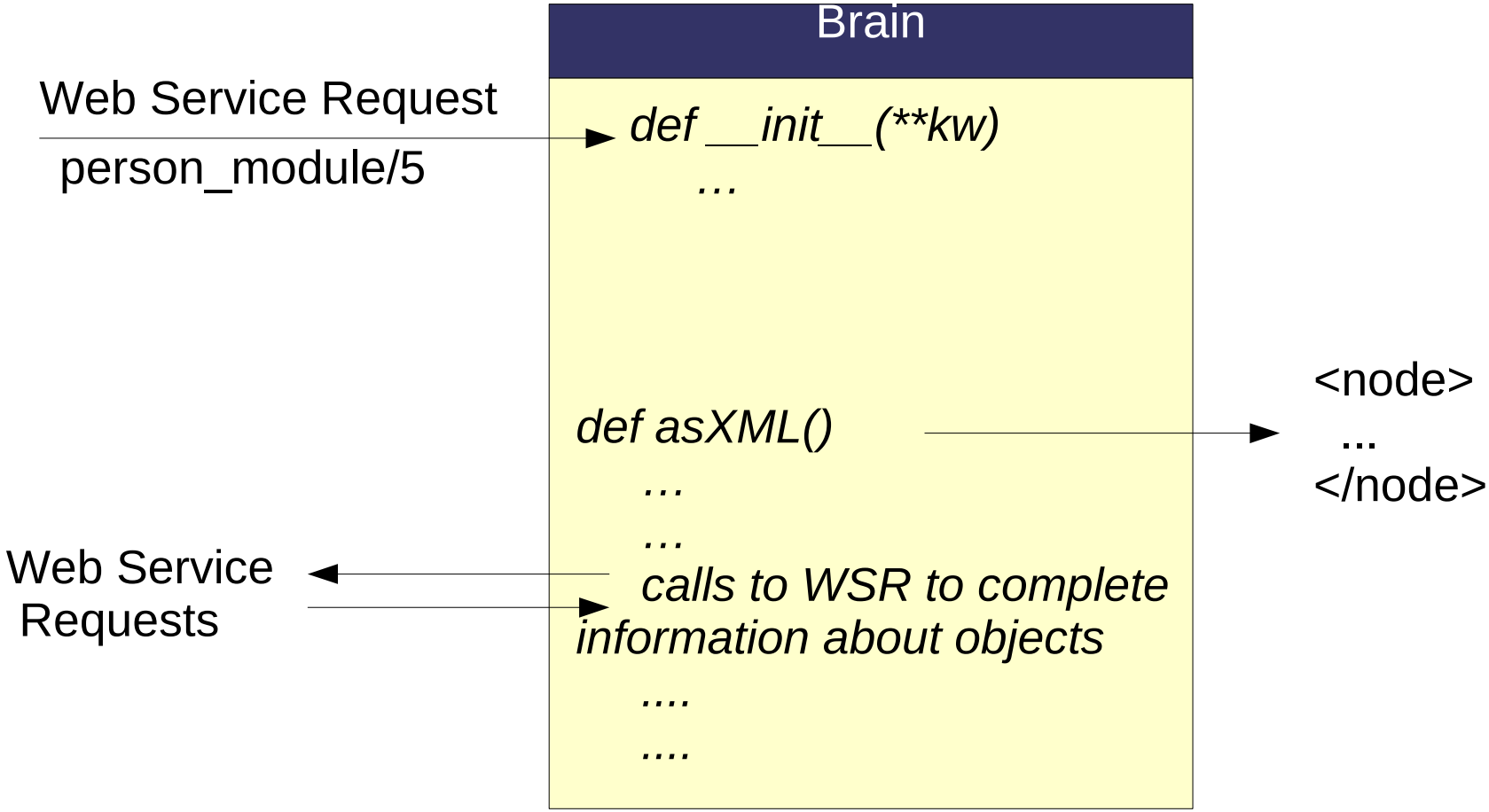
lxml

```
kw = {
reference : bg-1234,
title : Bag }
```

Parameters give to create an ERP5 object

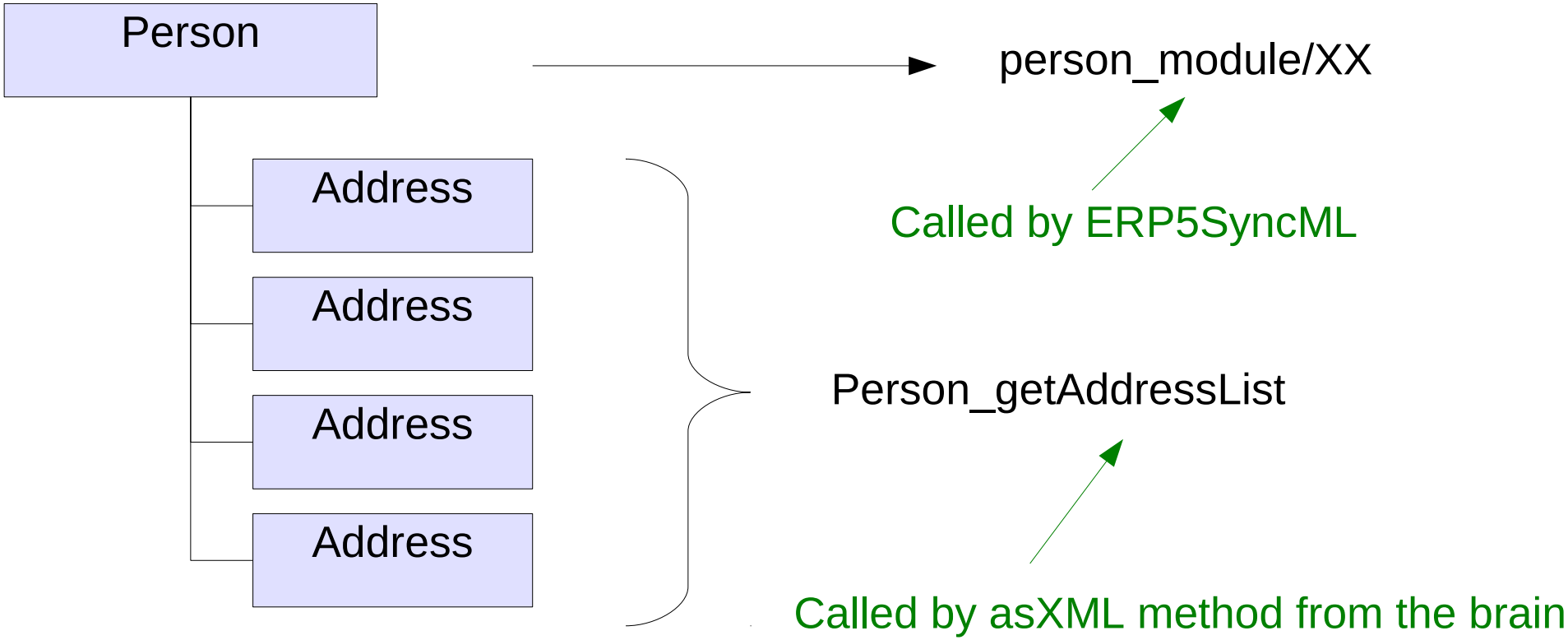
© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

XML Generation using many WSR



© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Example of Person



© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

How to test a Web Service Request

http://localhost:8080/erp5/portal_integrations/magento/person_module

Returns a list of Brain objects (1 brain = 1 person)

```
[<PersonWebServiceBrain object at 0x7ffb96387ed0>, <PersonWebServiceBrain object at 0x7ffb96387f50>, <PersonWebServiceBr
```

http://localhost:8080/erp5/portal_integrations/magento/person_module/8/asXML

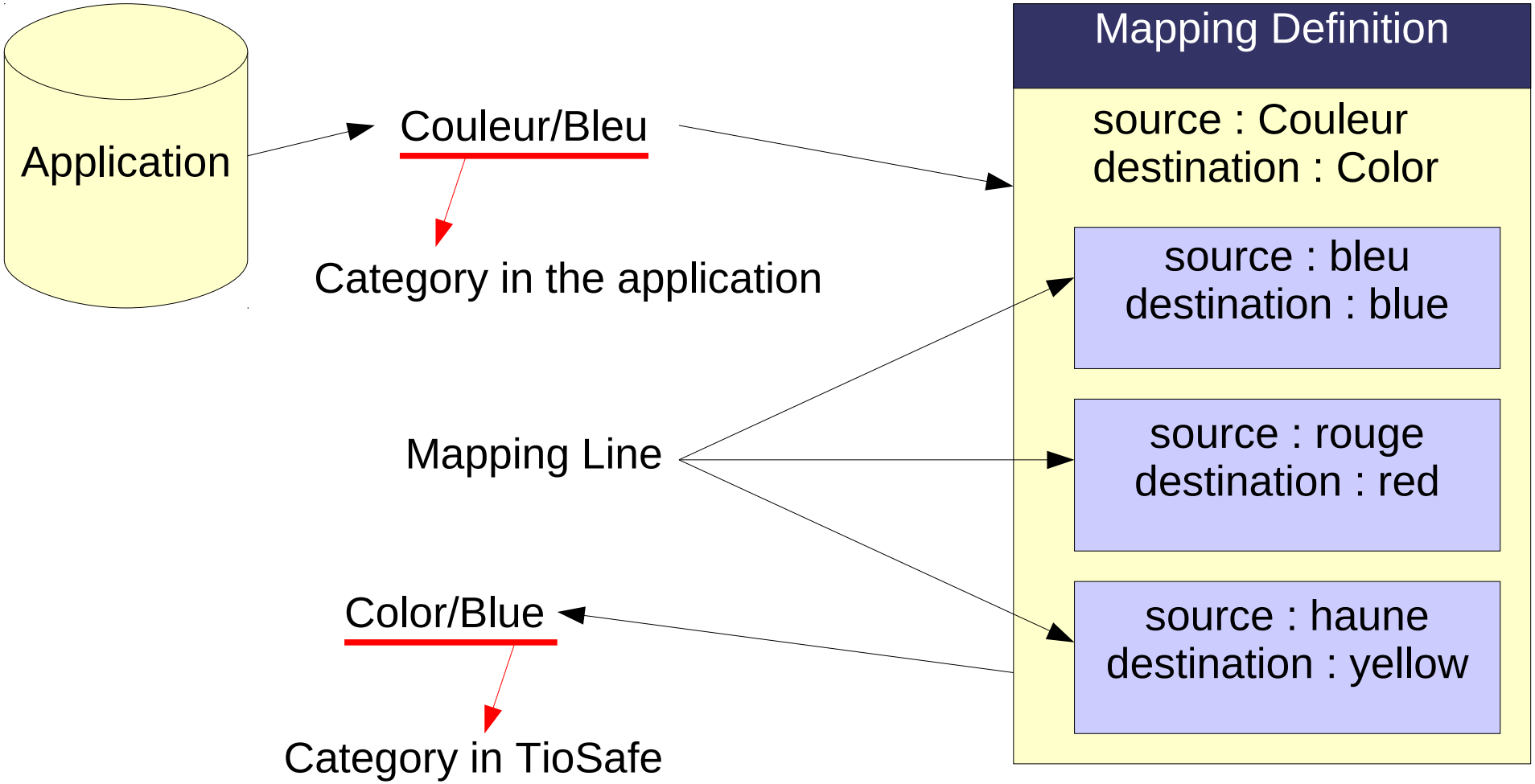
Local object ID in application (here magento)

```
<node>
  <title>John Doe</title>
  <firstname>John</firstname>
  <lastname>Doe</lastname>
  <email>john@doe.eu</email>
  <address>
    <street>8 rue du 14 juillet</street>
    <zip>59000</zip>
    <city>Lille</city>
    <country>France</country>
  </address>
</node>
```

Returns representation of the object in TioSafe XML

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Mapping



© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Synchronization Configuration

View | **Synchronization Configuration** | Mapping | History | Metadata

Synchronization Configurator Items: 3 records

Synchronization Order	Portal Type	Publication	Subscription
0	Person	magento_Person_pub_synchronization	magento_Person_sub_synchronization
1	Product	magento_Product_pub_synchronization	magento_Product_sub_synchronization
2	Sale Order	magento_SaleOrder_pub_synchronization	magento_SaleOrder_sub_synchronization

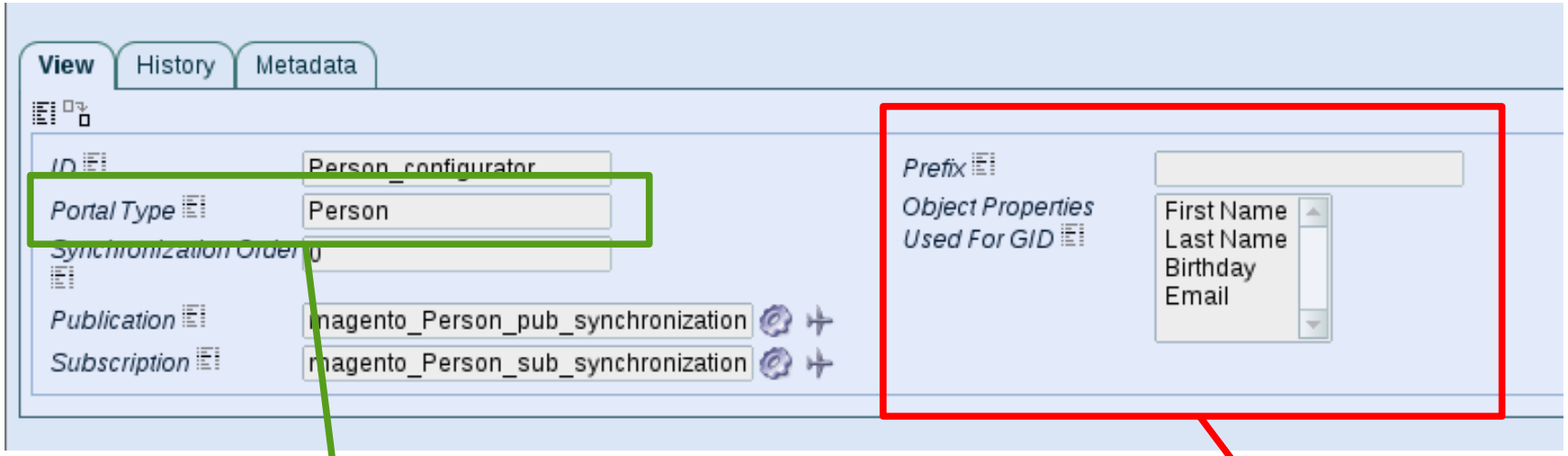
Define the order of synchronization. Important as for example sale orders required to have persons & products already synchronized in order to work

Relation to synchronization objects

One Configurator per module synchronized

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Synchronization Configurator Item

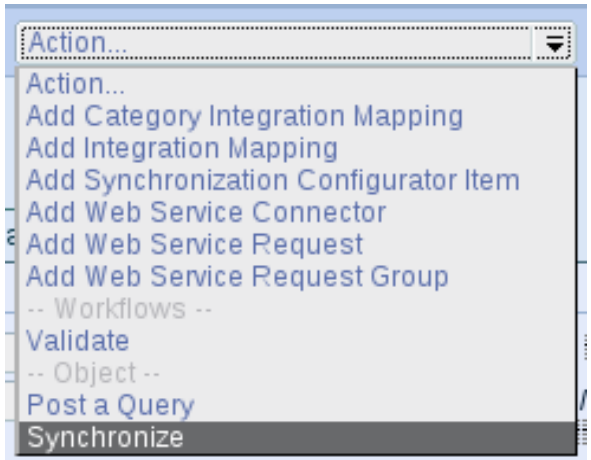


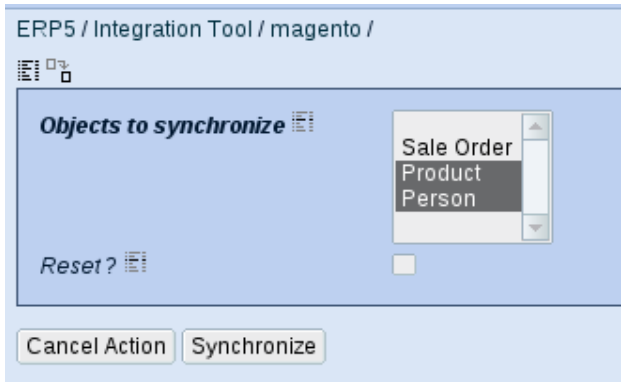
Type of object this configuration is applied to

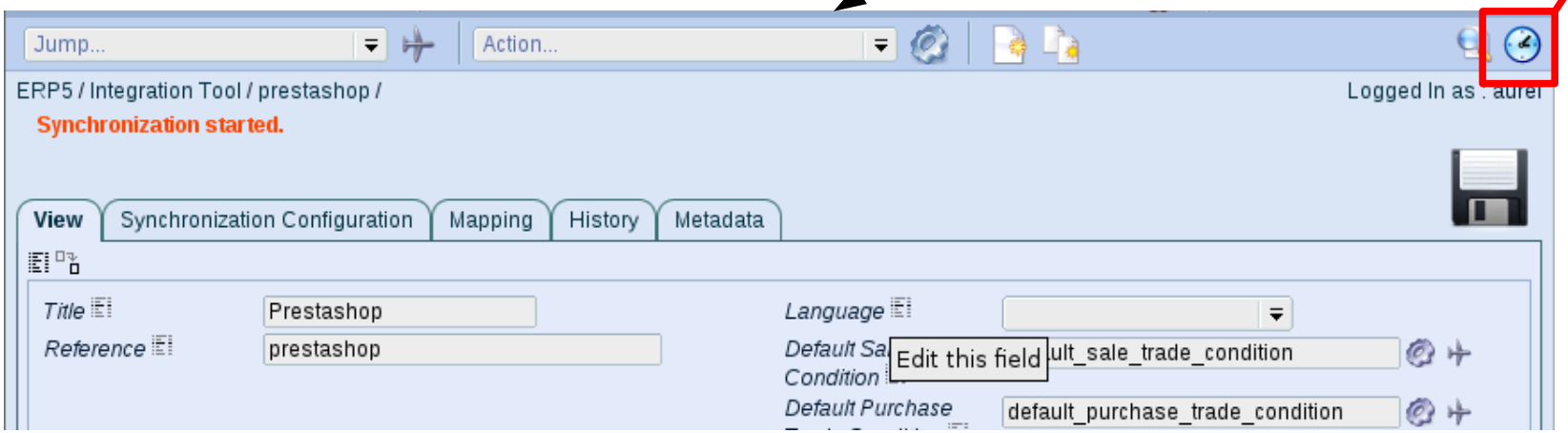
Define how GID is computed. Properties that can be used depends on module we synchronized

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Run a synchronization







Link to the view of background processes running for synchronization

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Status of Synchronization

Overview | Activities | LoadBalancing | Advanced | Contents | View | Properties | Security | Undo | Ownership | Find | History | Workflows

CMF Activity Tool at /erp5/portal_activities

Queued Activities

Warning: Invoking or canceling activities will invoke/cancel all activities with both the same path and the same method_id.

Warning: Invoking an activity will **not** run any dependency check.

	Uid	Object	Method Id	Activity Kw	Arguments	Named Parameters	Processing Node	Retry	Processing	Call Traceback
Cancel Invoke	2214 (dict)	/erp5/portal_synchronizations	SubSync	'tag': 'portal_integrations/prestashop', 'after_tag': []	'/erp5 /portal_synchronizations /ps_person_sub'		1	0	0	
Cancel Invoke	2215 (dict)	/erp5/portal_integrations /prestashop	getTitle	'after_tag': []			0	0	0	

Refresh pages clicking “Activities” tabs until list is empty
or activity in processing_node=-2 → this means there is an error

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Error view

URL :http://localhost:18080/erp5/error_log/manage_main

Log | Undo | Ownership

Site Error Log at /erp5/error_log

This page lists the exceptions that have occurred in this site recently. You can configure how many exceptions should be kept and whether the exceptions should be cop

Number of exceptions to keep 20

Copy exceptions to the event log ☑

Ignored exception types

```
NotFound
Redirect
```

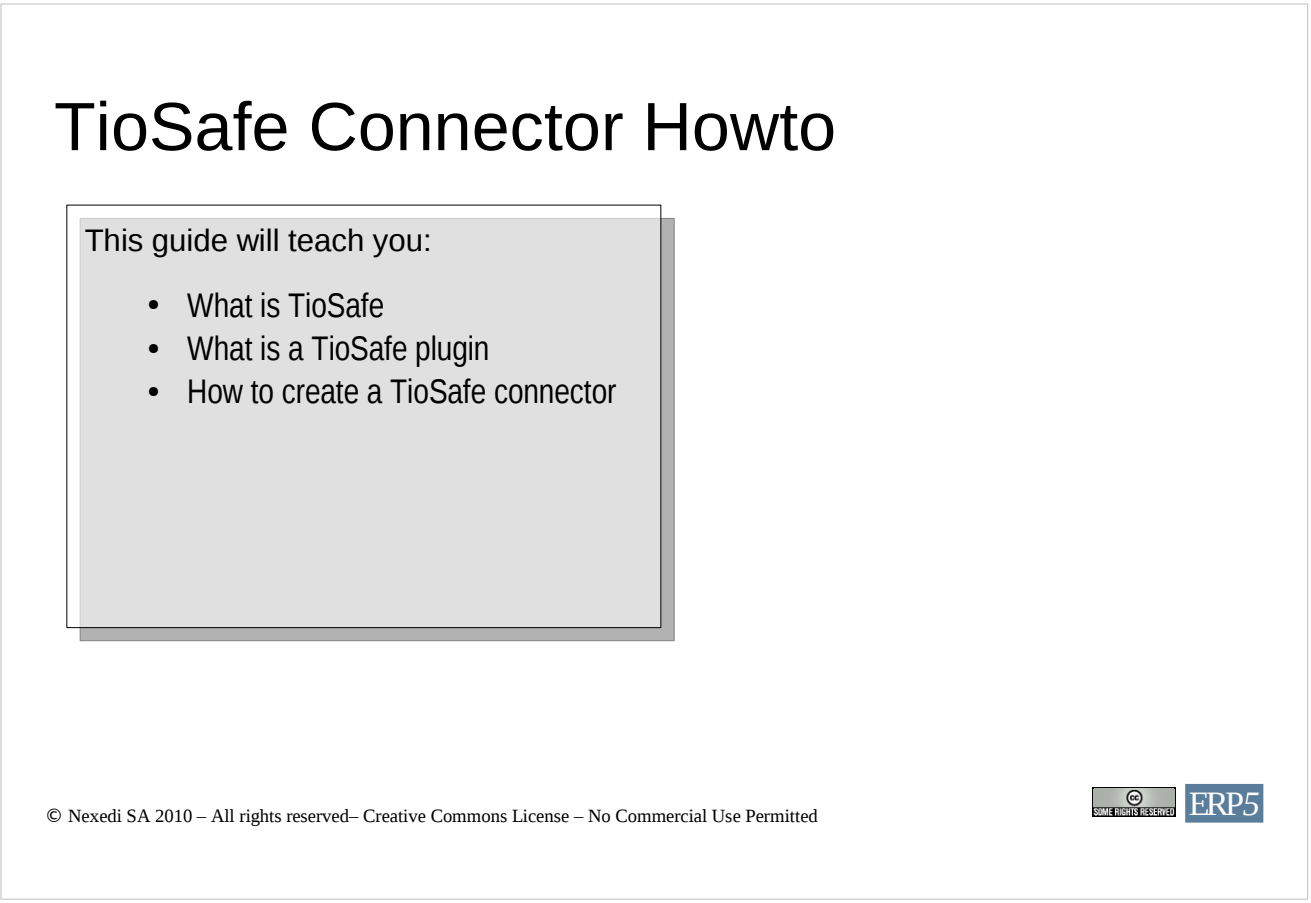
Save Changes

Exception Log (most recent first)

Time	Username (User Id)	Exception
16:17:32	Anonymous User (None)	AttributeError: 'NoneType' object has no attribute '__of__'
15:59:49	aurel (aurel)	ValueError: Bad XML returned by request /erp5/portal_integrations/prestashop/sale_...
15:57:49	aurel (aurel)	TypeError: unsupported operand type(s) for +: 'NoneType' and 'str'
15:57:43	Anonymous User (None)	Unauthorized: You are not allowed to access 'prestashop' in this context

Refresh

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted



This visual guide has been created for learning and for teaching ERP theory and practice, in association with TioLive ERP5 based plaform and the OSOE (One Student One ERP) programme.

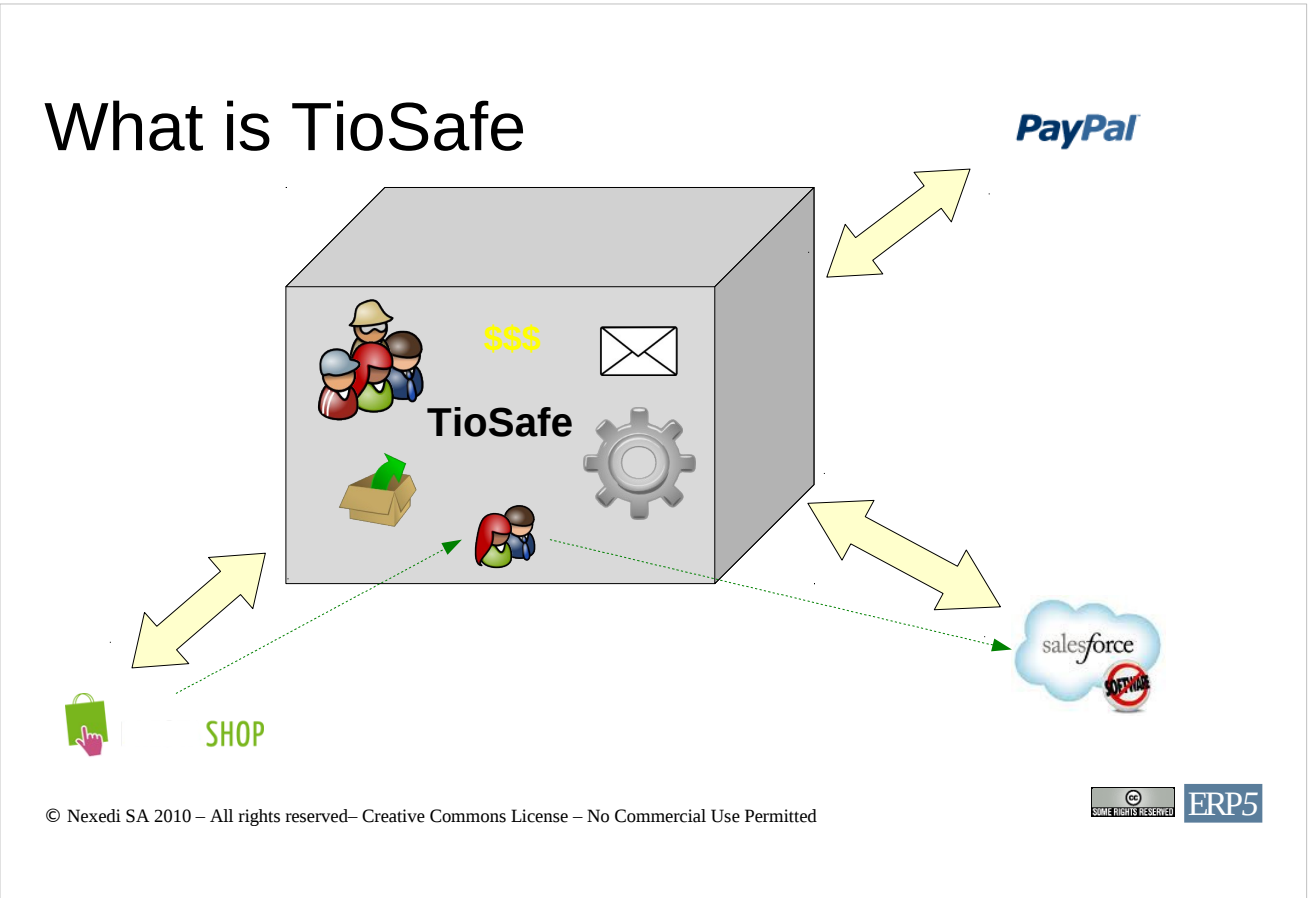
For learning ERP5 alone, readers should first have a quick look at the illustration on the upper part of the page then read the text bellow carefully and associate each word written in bold to the corresponding item(s) in the illustration. For example, the term **Creative Commons License** is written in bold because it defines the license of the above illustration.

This visual guide can also be used as a presentation for teaching ERP theory and ERP5 to a public audience. In this case, teachers are required to request an educational or a commercial license to Nexedi SA.

Copyright

You are free to copy, distribute, display, and perform the work under the following conditions: you must attribute the work in the manner specified by the author or licensor; you may not use this work for any commercial purposes includind training, consulting, advertising, self-advertising, publishing, etc.; you may not alter, transform, or build upon this work.

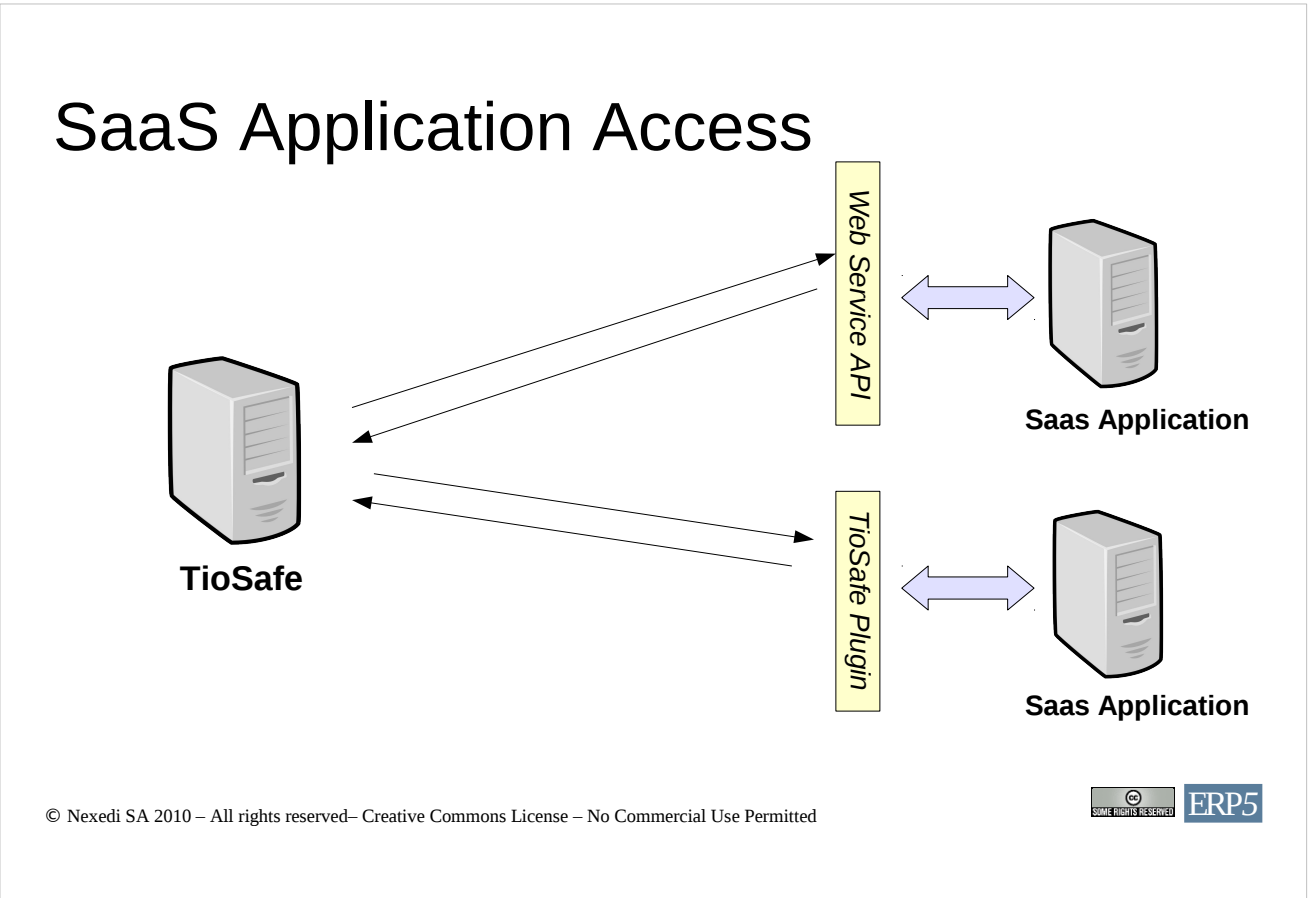
For any reuse or distribution, you must make clear to others the license terms of this work. Any of these conditions can be waived if you get permission from the copyright holder through a commercial license or an educational license. For more information, contact info@nexedi.com



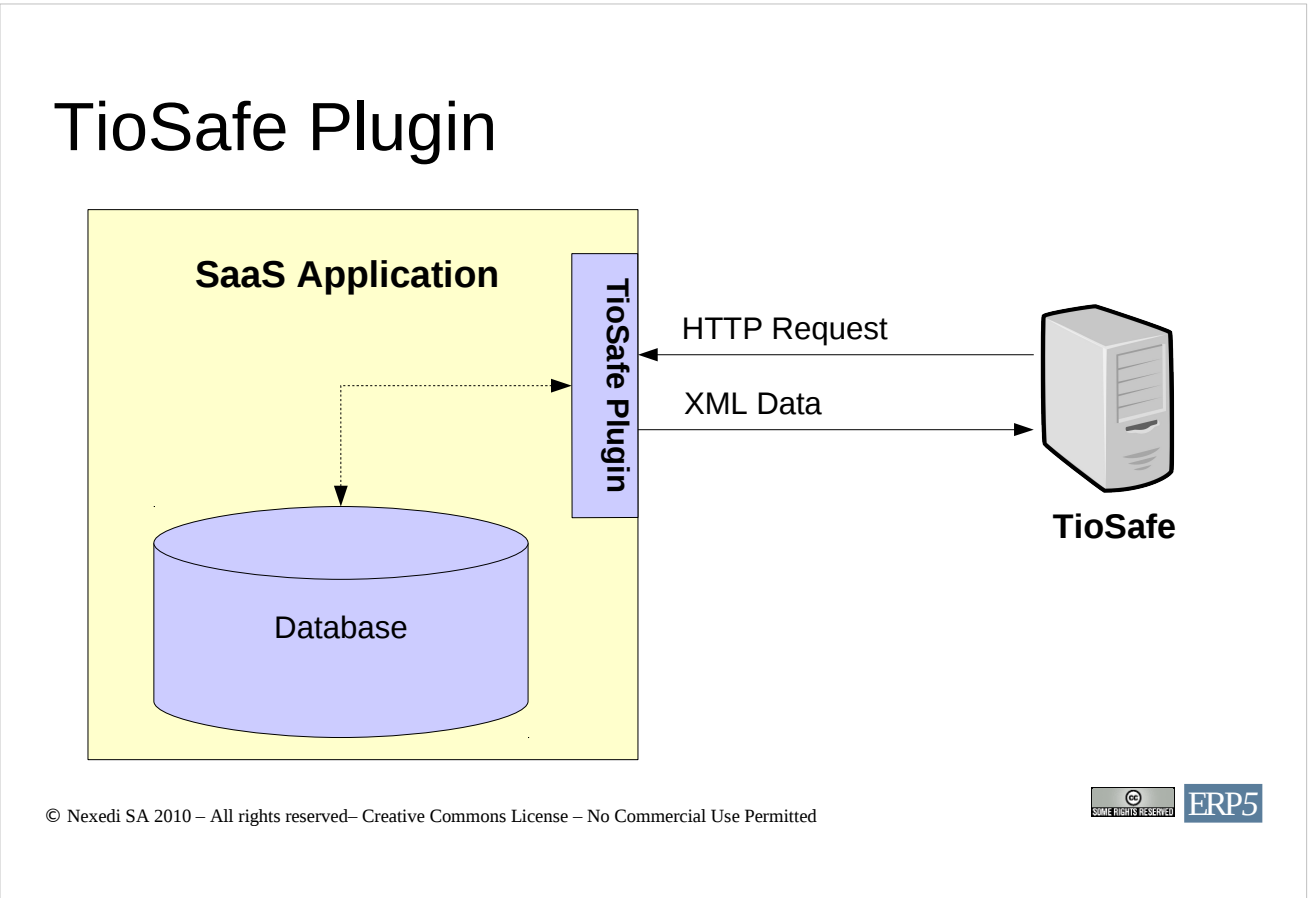
TioSafe is a platform which allow you to synchronize SaaS applications in between them. Thanks to TioSafe, integration of your data between many Saas is now easy.

For example, with TioSafe you can retrieve your clients from your e-commerce website under Prestashop and put them into your CRM software like SalesForce. Another example : if you no longer want to use SalesForce, TioSafe can synchronize all your events, campaign from SalesForce to SugarCRM. TioSafe makes you no longer dependant of the service your are using. And if you do not want to leave your data in SalesForce. TioSafe can also scramble all your data in the no longer desire application.

Another functionnality of TioSafe is that it acts as a sanctuary, keeping all your data in case you can no longer access them from other Saas.



There is two way for TioSafe to access other applications'data. Either the application provides a web service, in this case TioSafe directly use it. Either the application supports plugin. In this case a specific plugin has to be created, this plugin will then act as a web service TioSafe will question.



TioSafe plugin can be questionned by simple http request. they are used to work on data from the application. Usually developping a plugin is just creating the sql methods to question and update the application's database. Results of request must be return as XML, there is no specification for this XML as it will be transform to a “brain” object in TioSafe using a mapping dictionnary.

What a TioSafe plugin must do

- Provides access to data from application based on simple http request :
  - Get data
  - Put data
  - Update data
  - Delete data
- Supports authentification if not provided by the application
- Act as an oauth client regarding the TioSafe server

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Here is the list of what a TioSafe plugin must implements. Along the access to data, the plugin must make sure it can not be accessed anonymously, this should use the application framework if possible. Another requirement is that the plugin authenticates himself to the TioSafe server using the oauth protocol.

See documentation about Oauth :

- Oauth Website
- Oauth Guide
- Oauth example code

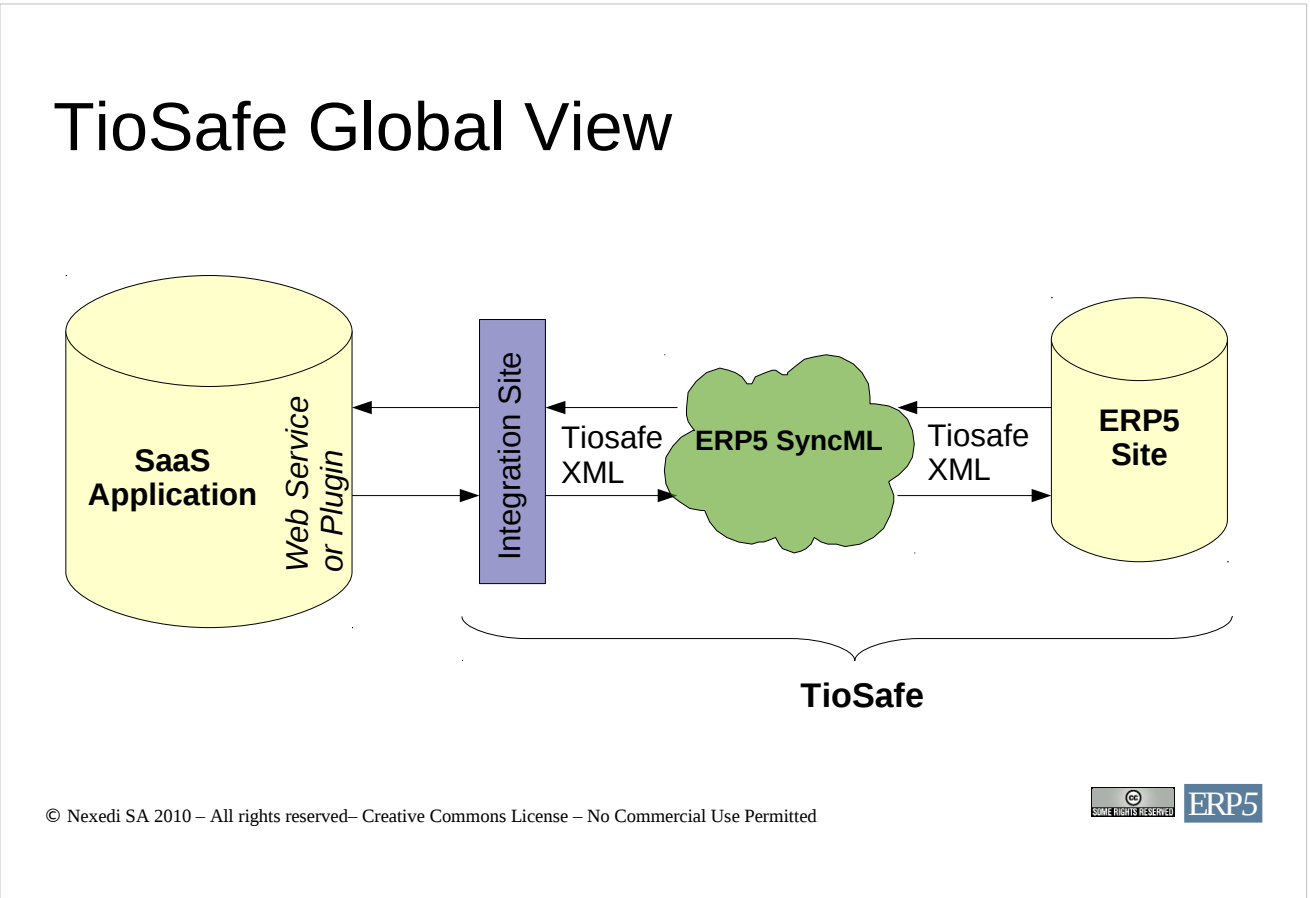
Which data must be retrieved

Data returns by plugin or web service calls depends on object type :

- Nodes : Persons, Organisations, Accounts
- Resources : Products, Services, Components
- Transactions : Orders, Events, Accounting transactions, Packing Lists

All properties exchanged are defined in XSD files contains in the ERP5TioSafe products

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted



This is the global view of how a SaaS Application and TioSafe communicate. Access to the SaaS Application is done by an “Integration Site” (this is what we call the connector). An “Integration Site” defines how to connect to the application (protocol, url, identification...) and all methods to call in order to question or update data in the application.

The “Integration Site” is then considered as an ERP5 Site so that it can be synchronized into a real ERP5 Site using the SyncML Protocol. This ERP5 Site is where all data from all applications will be synchronized in the end, and thus it will be used to synchronized data from an application to another one.

The development of a “Connector” consists in creating and configuring an “Integration Site”.

Requirements

TioLive instance + ERP5TioSafe Product + erp5_tiosafe_core bt5

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

In order to develop your own integration site for your application, you need to have a TioLive instance

- TioLive

Visit this link to get details about installation & configuration :

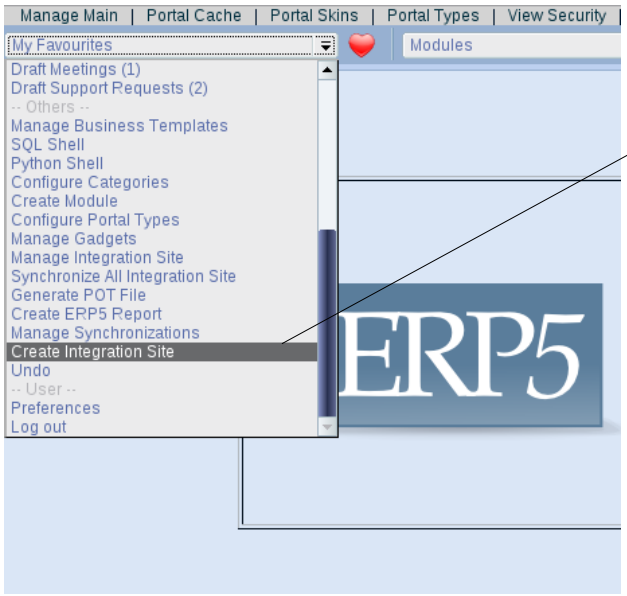
- Wiki ERP5

Once erp5 installed you need to ass specific products & business templates for TioSafe that can be found there :

- https://svn.erp5.org/repos/tiosafe

At least the “erp5_tiosafe_core” must be installed, other bt5 are example of integration sites for specific applications.

Creating an Integration Site



This will redirect to the form used to create a new integration site

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Once everything is installed, you can access the list of integration sites using the menu “Favourites” and the link “Manage Integration Sites”

Integration Site Creation Form

Type of connections used to access the Web Service or Plugin

Lists of modules that will be synchronized

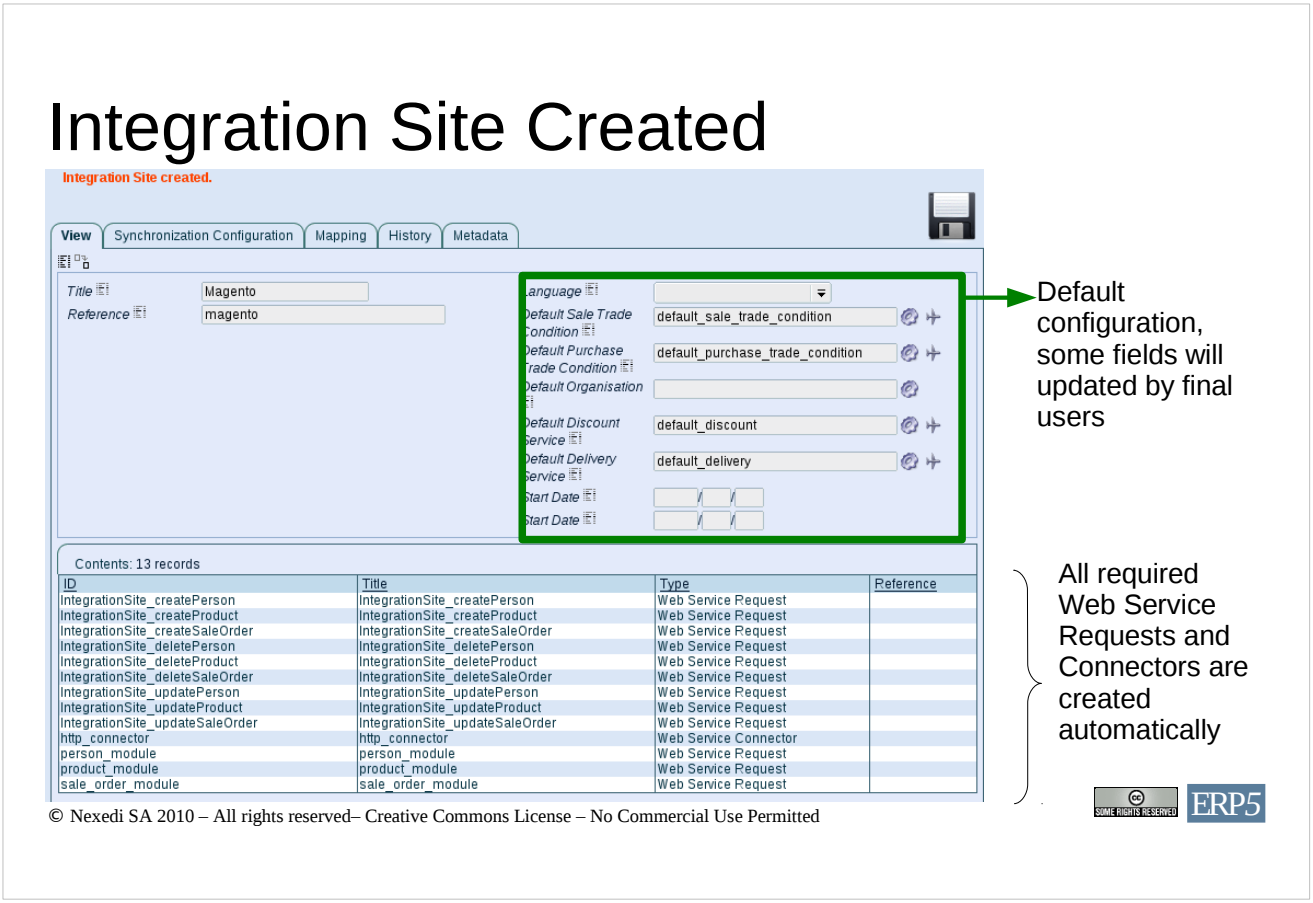
© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

This form will help you create a pre configured integration site. Here is the details about each parameters:

- Application name : must be unique, usually name of the application is sufficient

- modules to synchronize : you must select the list of modules that will be synchronized with your applications. For example, in case of an e-commerce website, we will synchronize people, products and sale order, so in this lists “Person”, “Products” and “Sale Orders” must be selected.

- connectors to use : you must select the kind of transport that will be used to exchange data between your application and TioSafe. If you develop a plugin by yourself in the application, you must use http.



The newly created Integration Site comes with :

- all required web service requests and connectors objects

- default configuration for synchronization process

- default configuration for each synchronized module (tab “Synchronization Configuration”)

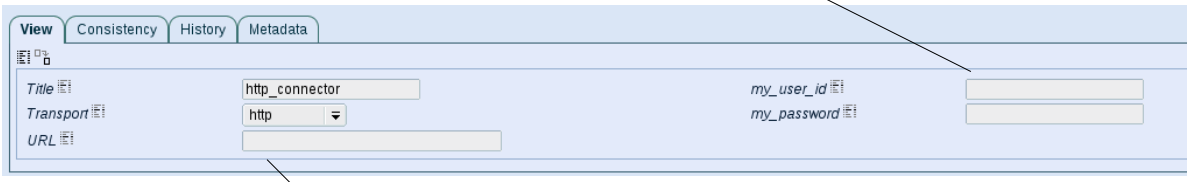
What you must define/create :

- Default organisation : this is the organisation representing the user which is synchronizing its application, for example “Shop XXX” as we are synchronizing e-commerce application of “Shop XXX”. Define your own organisation here, note that it might be synchronized in the application is you are synchronizing the organisation module.

Clicking on a line will bring you to the sub-object, on this screen either a Web Service Request or Connector.

Web Service Connector

Authentification parameters

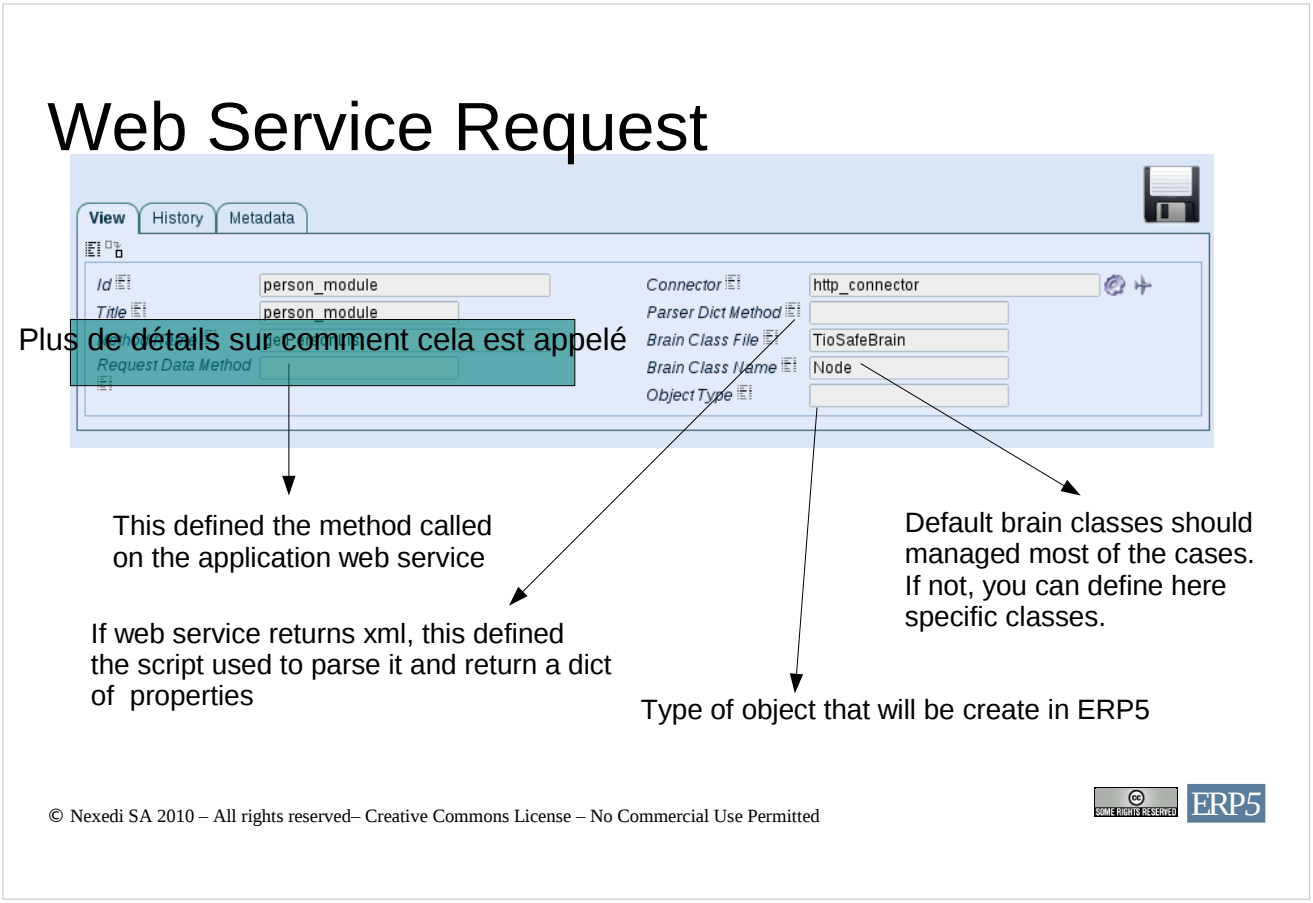


URL which provides the service
(must be a complete url that include the plugin path)

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

To use the web service, you need to fill the url of your web service and authentification informations.

URL of the web service must be complete, this means that it might be your plugin path for example : http://myapplication.com/administration/modules/oneclickconnect/



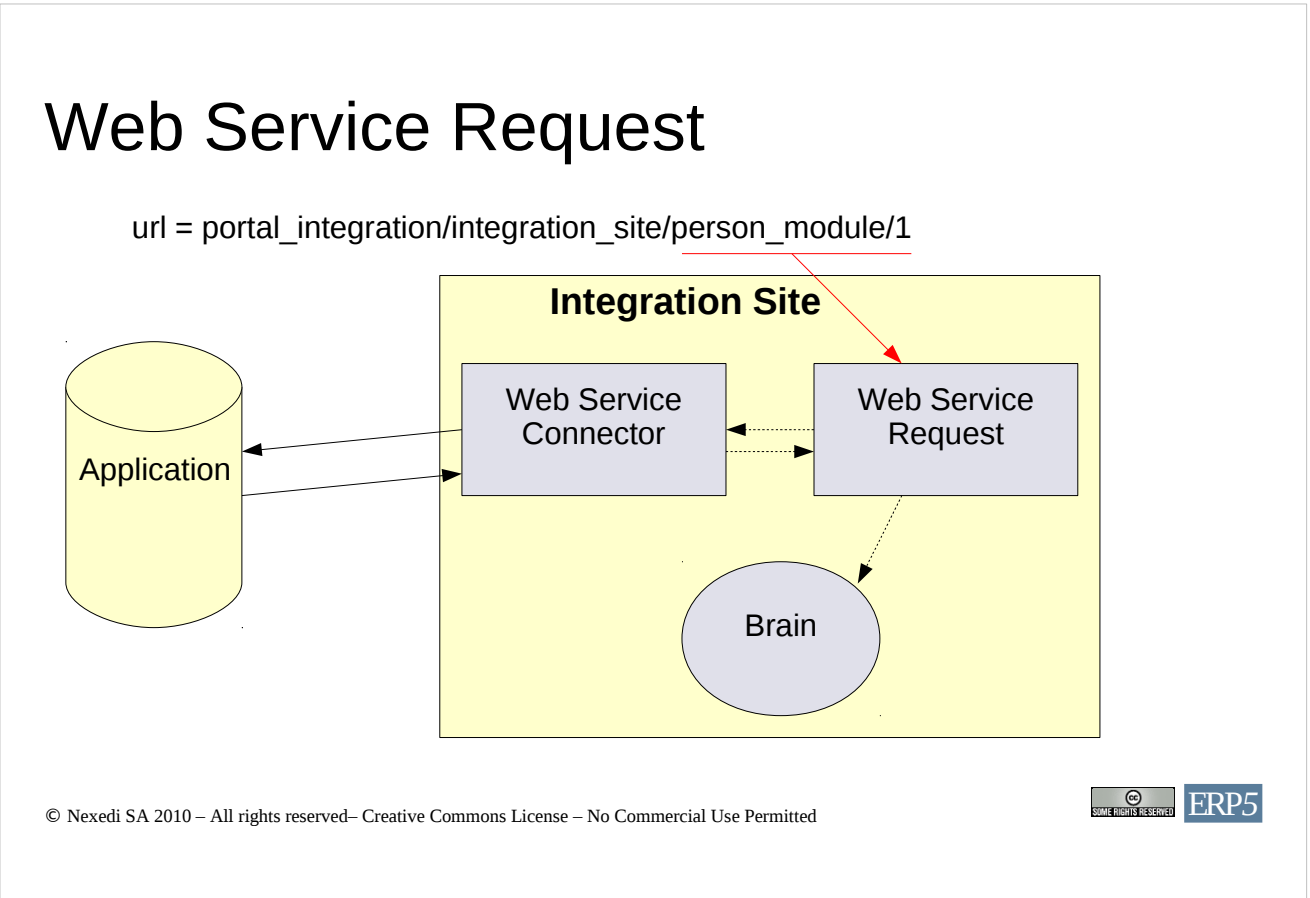
Web Service Requests are created with defaults parameters. Each one must be completed to define at least the method that will be called on the application's web service (Method Name field).

Request Data method is used for web services that required a request body like SOAP requests for example. In this case, a Page Template is created to define the body and the name of it is defined in this field. (See following slides for an example)

Brain class are used by methods which returns objects. There is some default classes which should manage most of the cases. If specific cases happens, this allow to use a custom class. Creation of a specific brain will be required for transactions objects for example due to the link with other object of the application. This part is treated later in the documentation.

Parser dict method is used for web services that returns XML data. As this XML is not expected to follow any specification, it must be converted to ERP5 object properties. This field allow to defined a script to call that will return the mapping dictionnary used by the sax parser to do this mapping. (See following slides for an example).

Object Type defined the type of object that will be created on ERP5 side, this is used only by *_module methods.



This is simple case to retrieve data from an application. When we want an object (or a list of object), only one request is done. Web Service Request defined what methods must be called and how to interpret its result. Web Service Connector will do the connection to the application, emits the request, get the result and return it to the Web Service Request. The result is then parse, and a brain object is created. The brain object simulates an erp5 object and defined the asXML method which generates the XML used for the synchronization. The XML format returned by the asXML is based on the TioSafe XML Specification. Note that the brain might need to do some requests to the application on order to complete the XML. (For example to retrieve all addresses of one person)

How to define ID of WSR

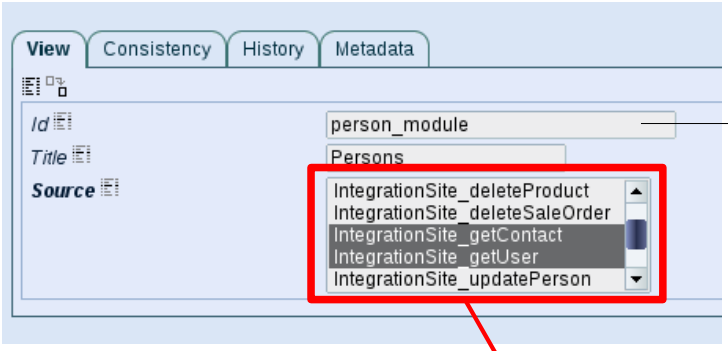
portal-type : Type of the object in TioSafe

- *portal-type*_module : simulate a TioSafe module, used to retrieve object
- IntegrationSite_create*Portal-Type* : method called to create object in the application
- IntegrationSite_update*Portal-Type* : method called to update object in the application
- IntegrationSite_delete*Portal-Type* : method called to delete object in the application
- IntegrationSite_get*Portal-Type*LastID : method called to get the last ID in the application for Portal-Type

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

ERP5SyncML need to retrieve/add/update/delete objects from the application. We thus need to configure all methods in the integration site for each kind of object that will be synchronized. Note that this is the minimal methods required, you might need other to get specific data (categories of product, addresses...). All methods must follow the scheme IntegrationSite_get*Portal-Type*Something except for the one simulating TioSafe modules.

Web Service Request Group



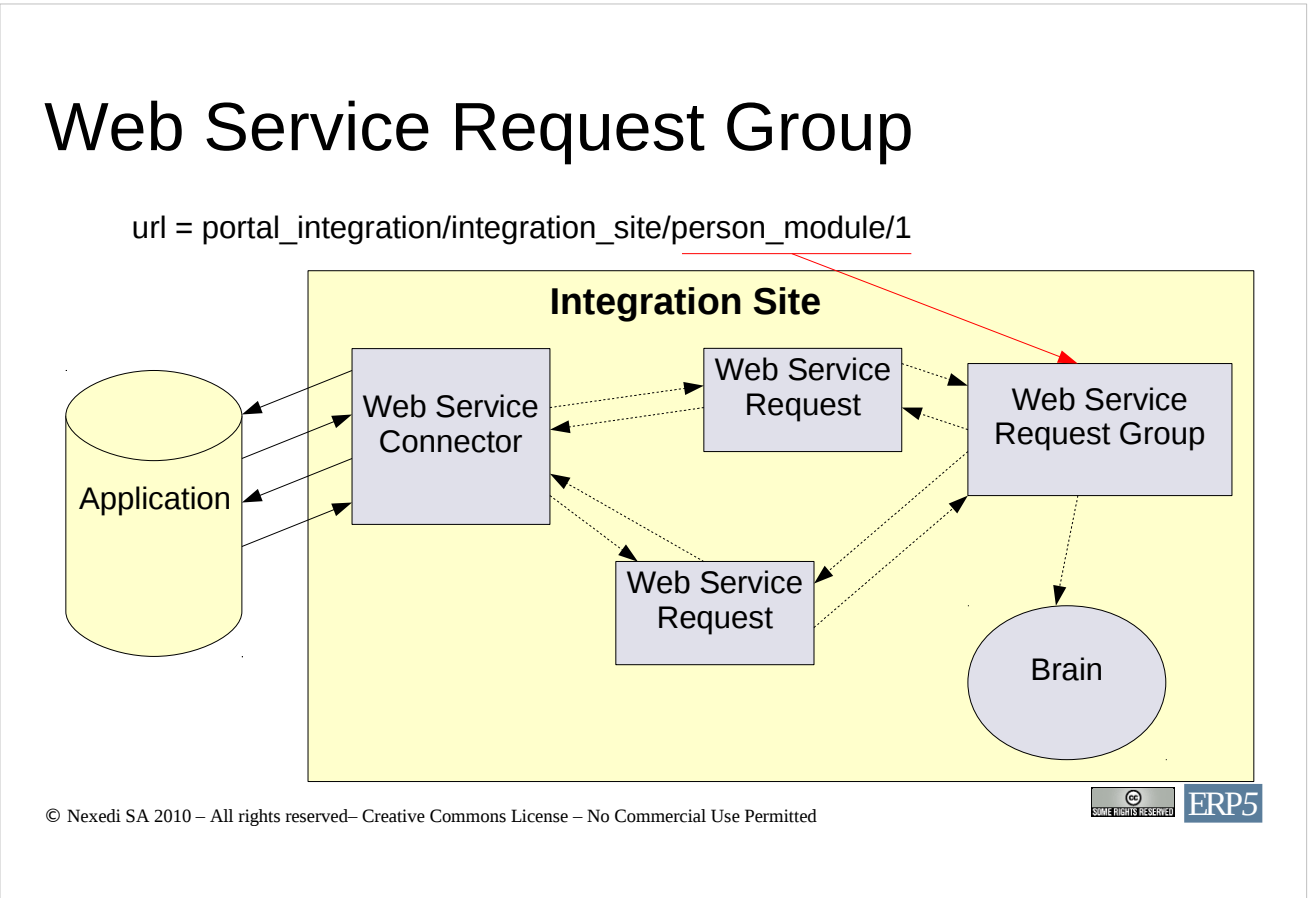
ID of the method called by TioSafe

List of Web Service Requests used

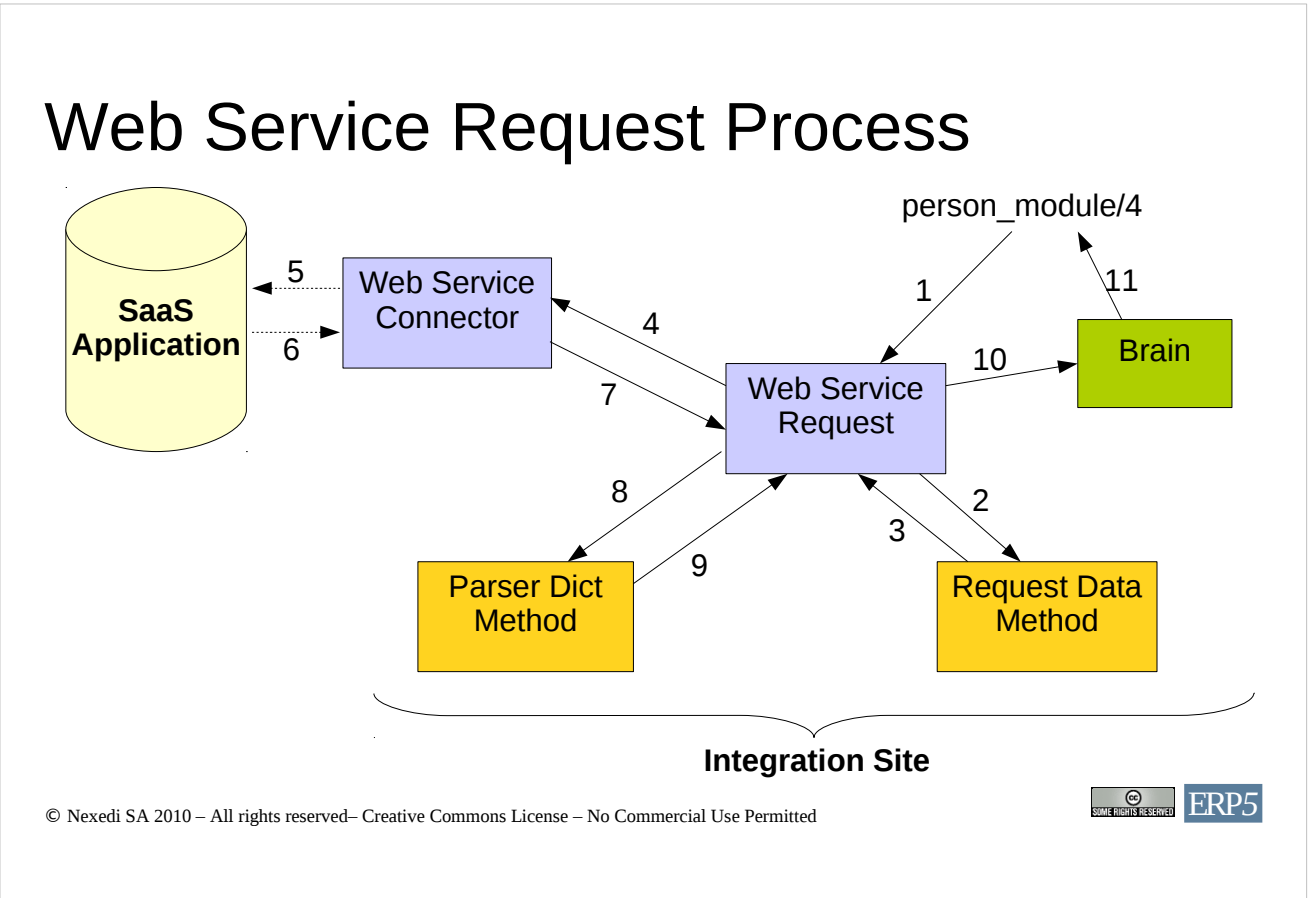
© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Web Service Groups are used for cases where multiple requests must be emitted to retrieve/put data. For example, in order to get all persons, one might need to get all contacts (1 WSR) plus all users (1 WSR).

You can add a Web Service Group using the “Action” menu on the Integration Site. Web Service Request Group required a specific ID (as for Web Service Request), and you just have to select the Web Service Requests that it will used.



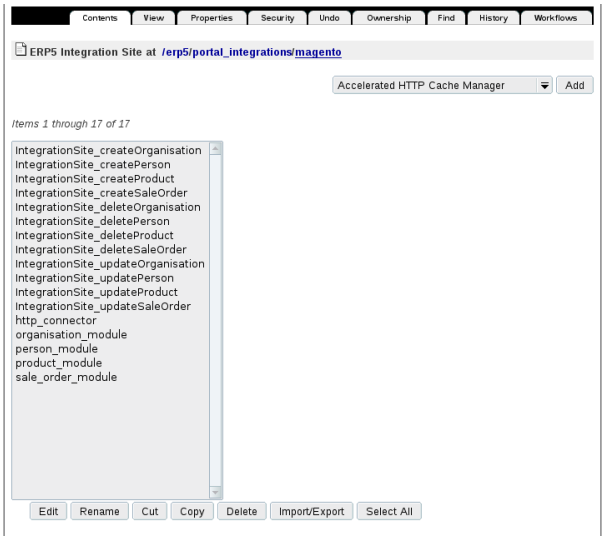
The process is the same as for a simple request, we just do it many times. Results are then aggregrated by the Web Service Request Group.



This slide describes the workflow done when TioSafe try to access application's data. First part consists in building the data of the request which is done with the “Request Data Method”, it will usualy be a page template used to format parameters (this is used in case of SOAP for example, but this is not required at all). Then the request is sent to application through the “Web Service Connector”. Once the web service request receives the answer, it calls the Parser Dict Method in order to transform custom XML into object properties which will be used to create a brain that will be return to the initial caller. Note that there is no answer in case of Web Service Request that put/update/delete objects in the application.

Access to the manager view

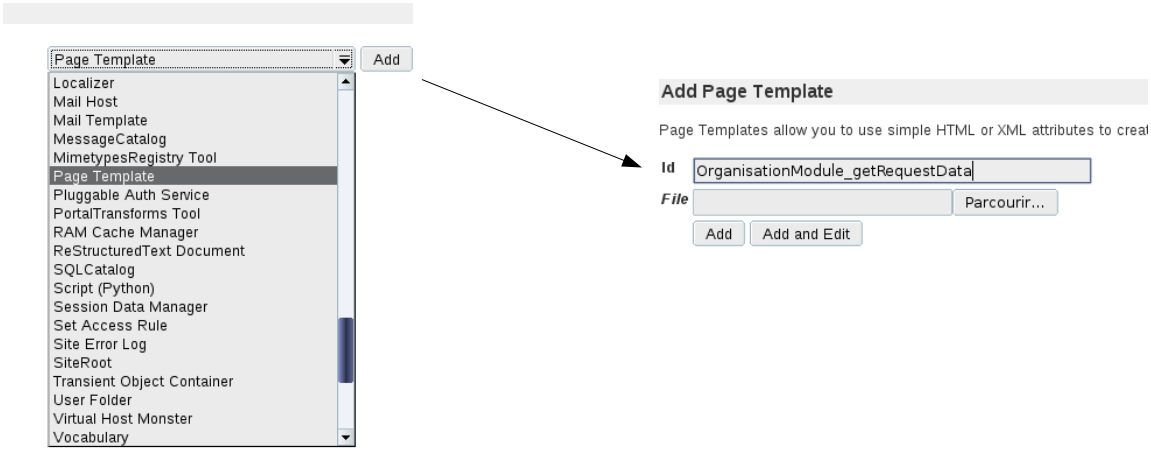
http://localhost:8080/erp5/portal_integrations/magento/manage_main



© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

In order to create Request Data Method or Parser Dict Method, you must use the manager view. To access it, you just need to add “manage_main” after the url of the integration site. You will then see the list of all already created sub objects. There is also a list which is used to add new object.

Request Data Method Add



© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

To add a Request Data Method, browse the list and select “Page Template”. ID of request data method must be defined this way :

if WSR ID = organisation_module, ID is OrganisationModule_getRequestData

if WSR ID = IntegrationSite_updateOrganisation, ID is UpdateOrganisation_getRequestData

Request Data Method Edit

Page Template at /erp5/portal_integrations/magento/OrganisationModule_getRequestData

Title

Content-Type text/html

Last Modified 2010-07-01 03:04 PM

Browse HTML source

Expand macros when editing

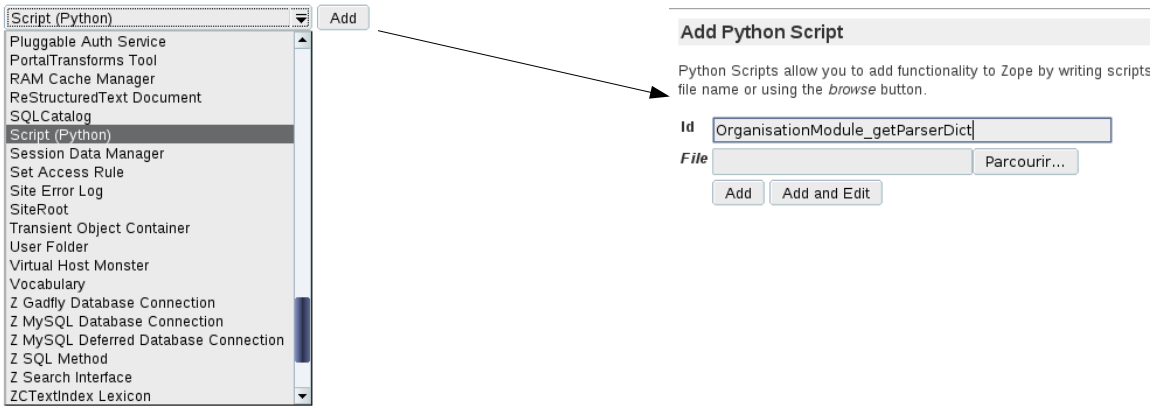
```
<ns0:query>
  <ns0:queryString>
    SELECT
      Id, Name, FirstName, LastName, Email, Phone, MobilePhone, Birthdate
    FROM
      Contact
    WHERE
      <tal:block tal:condition="python: options['id'] is not None">Id = '<tal:block tal:content="options/id" />' AND</tal:block> Email != NULL
  </ns0:queryString>
</ns0:query>
```

Example of body of a SOAP Request

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Once created you can define the body of the request. For more details about Page Template, check the Zope Page

Parser Dict Method Add



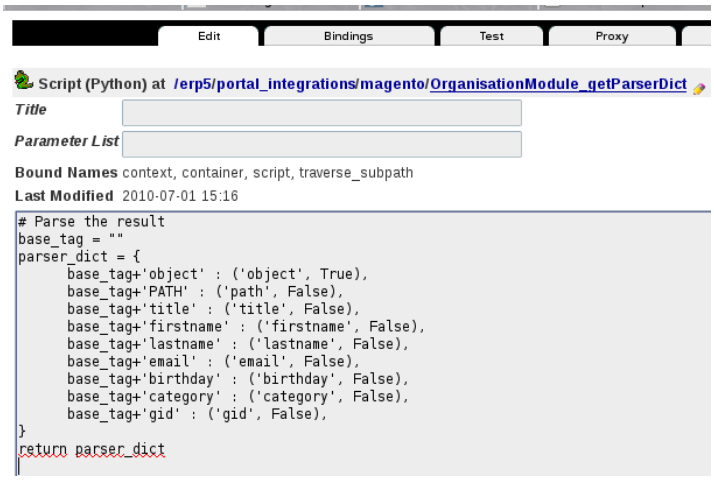
© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

To add a Parser Dict Method, browse the list and select “Script (Python)”. ID of parser dict method must be defined this way :

if WSR ID = organisation_module, ID is OrganisationModule_getParserDict

if WSR ID = IntegrationSite_updateOrganisation, ID is UpdateOrganisation_getParserDict

Parser Dict Method Edit

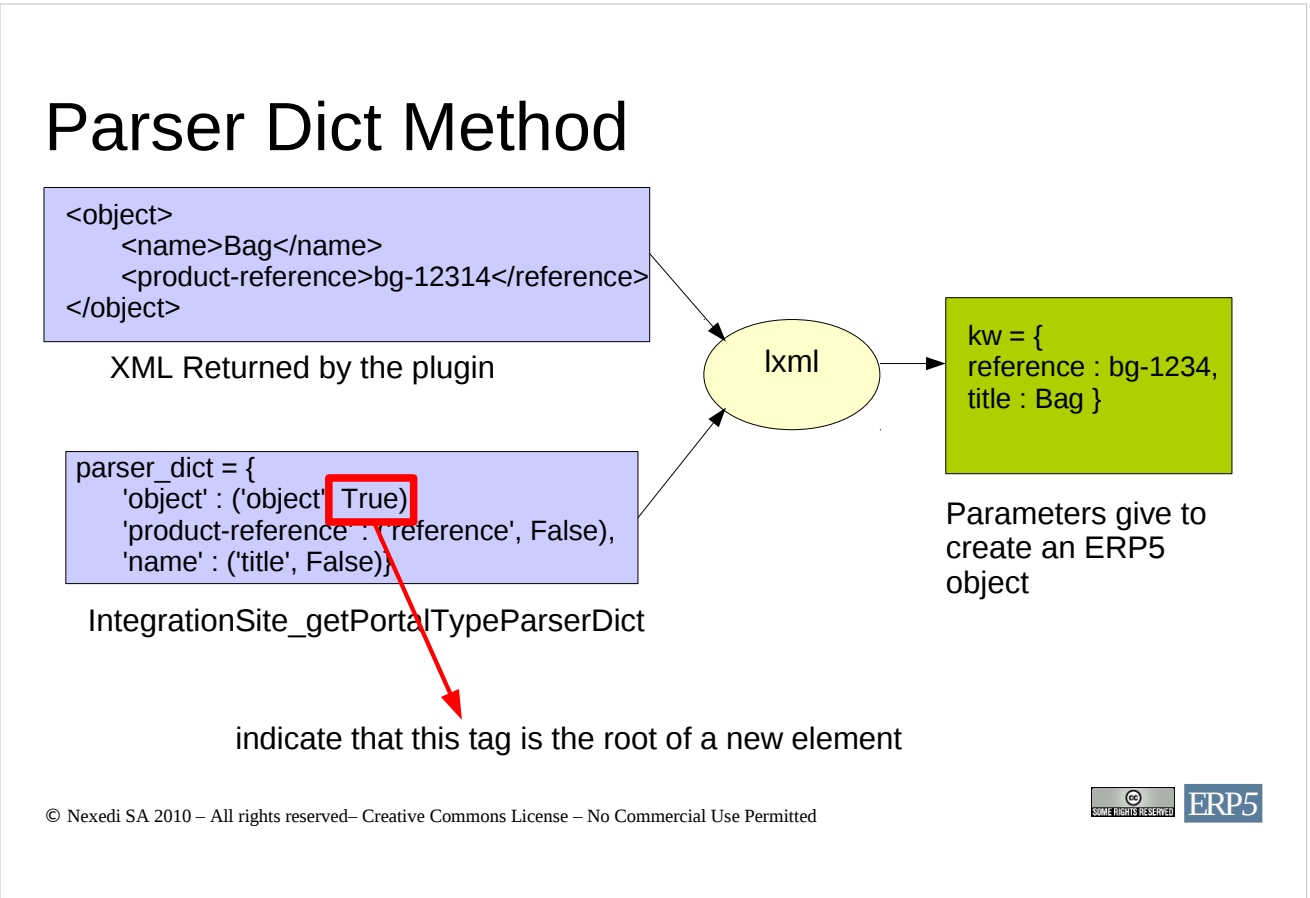


© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

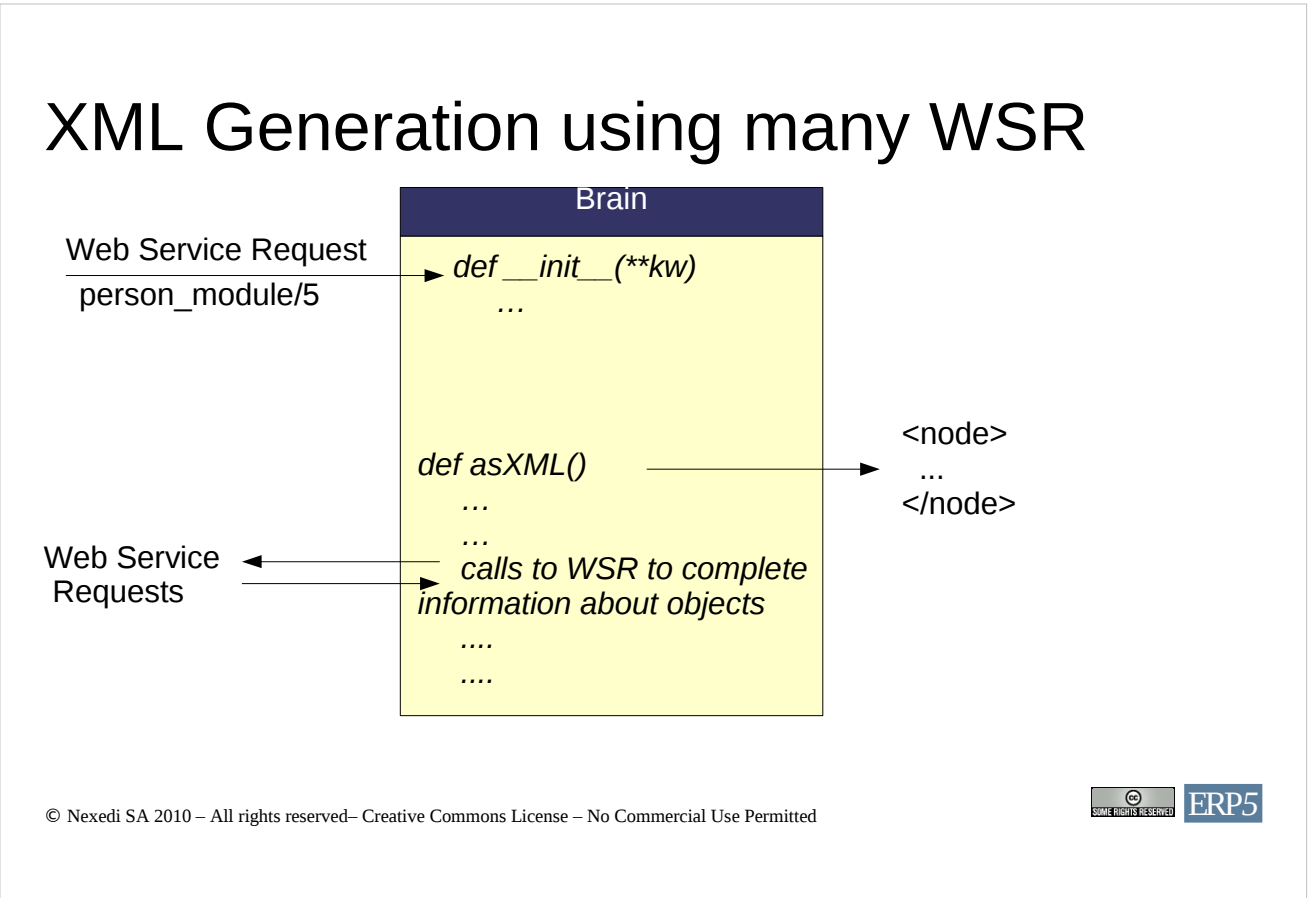
Once added, you can defined the dict that will be returned.

The structure of the dict must be :

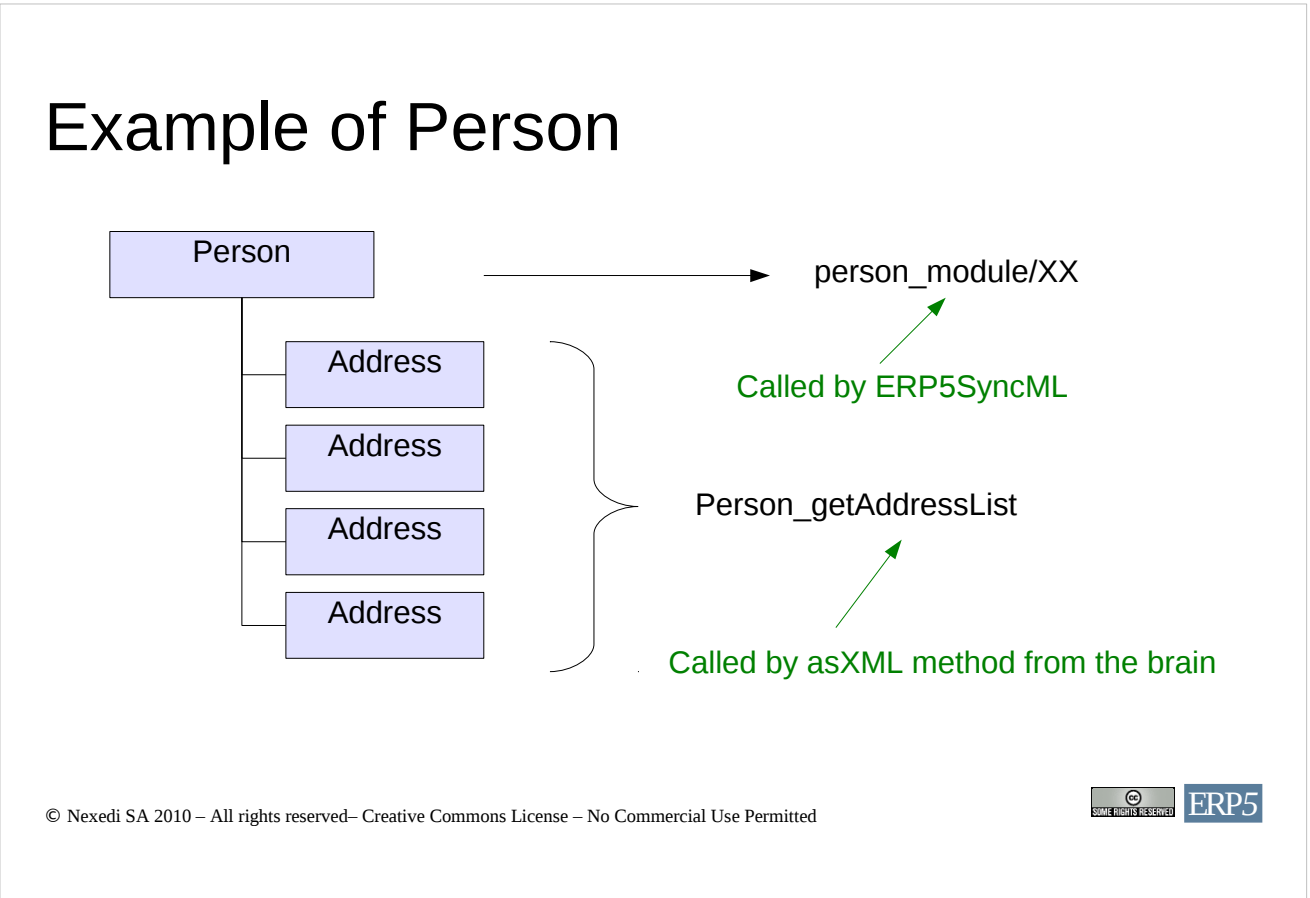
{ application xml tag : (erp5 property, boolean which indicate if root of a new element) }



Parser dict methods are used to transform XML returned by web service into a dict which is used by ERP5 to create brains objects which simulates an ERP5 object. As name of tags used by a web service can be different, a mapping between this tags and ERP5 object properties has to be done. This is achieved thanks to the Target Parser Interface available in the lxml lib.



All Web Service Request automatically created are used by TioSafe. But we may need other web service request to complete information an object. In this case these Web Service Requests are called from the asXML method. As asXML method generates the complete XML view of an object, it is required to retrieve missing data that can't have been retrieve from the original WSR.



This is an example where many requests are required to get all data for one object. In case of persons, plugin or Web service can not return all the address at the same time as the information about the person. In this case, addresses will be retrieved when the asXML method called.

Here are some other cases :

- order lines of a sale order

- categories of a product

Note that it depends on the Web Service, one can give all data at a time where others can not.

ID of these request also follow some rules, construction of the ID is the following

*ObjectType*_get*SubObjectType/Property*List

Currently here the list of WSR used that way :

• Person_getAddresObjectList

• SaleOrder_getSaleOrderLineList

• Product_getCategoryList

How to test a Web Service Request

http://localhost:8080/erp5/portal_integrations/magento/person_module

Returns a list of Brain objects (1 brain = 1 person)

```
[<PersonWebServiceBrain object at 0x7ffb96387ed0>, <PersonWebServiceBrain object at 0x7ffb96387f50>, <PersonWebServiceBr
```

http://localhost:8080/erp5/portal_integrations/magento/person_module/8/asXML

Local object ID in application (here magento)

```
<node>
  <title>John Doe</title>
  <firstname>John</firstname>
  <lastname>Doe</lastname>
  <email>john@doe.eu</email>
  <address>
    <street>8 rue du 14 juillet</street>
    <zip>59000</zip>
    <city>Lille</city>
    <country>France</country>
  </address>
</node>
```

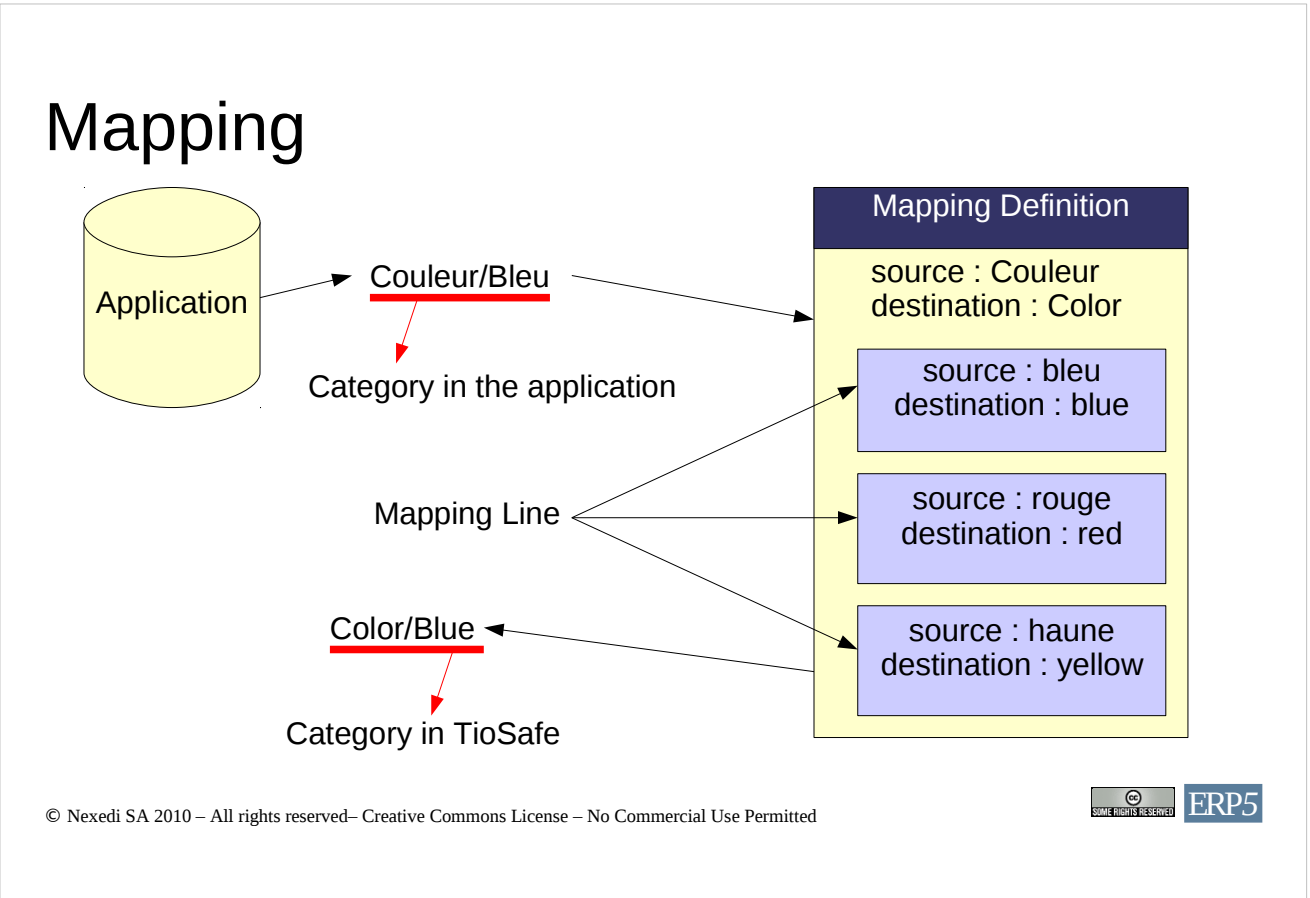
Returns representation of the object in TioSafe XML

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Web Services Request can be simply tested by calling their URL.

In case of module, the url (like *site/person_module*) must returns a list of brain objects. This represents the listing of a module.

As WSR are traversable, it is possible to call directly for an object given its id (like *site/person_module/8)*, this returns just one brain object. Calling asXML object on it (*site/person_module/8/asXml)* will returns the representation of the object in TioSafe XML.



For each integration site a mapping must be defined between categories/properties of the Application and the categories/properties/object of TioSafe. To achieve this, we use for categories :

- Category Integration Mapping and Category Integration Mapping Line

and for other properties

- Integration Mapping and Integration Mapping Line

This mapping is performed when calling asXML method on the brain. At least all base category mapping must be defined by the user to synchronize data. Mapping Line are not mandatory, ressource can use individual variations for example.

Synchronization Configuration

View | **Synchronization Configuration** | Mapping | History | Metadata

Synchronization Configurator Items: 3 records

Synchronization Order	Portal Type	Publication	Subscription
0	Person	magento_Person_pub_synchronization	magento_Person_sub_synchronization
1	Product	magento_Product_pub_synchronization	magento_Product_sub_synchronization
2	Sale Order	magento_SaleOrder_pub_synchronization	magento_SaleOrder_sub_synchronization

Define the order of synchronization. Important as for example sale orders required to have persons & products already synchronized in order to work

Relation to synchronization objects

One Configurator per module synchronized

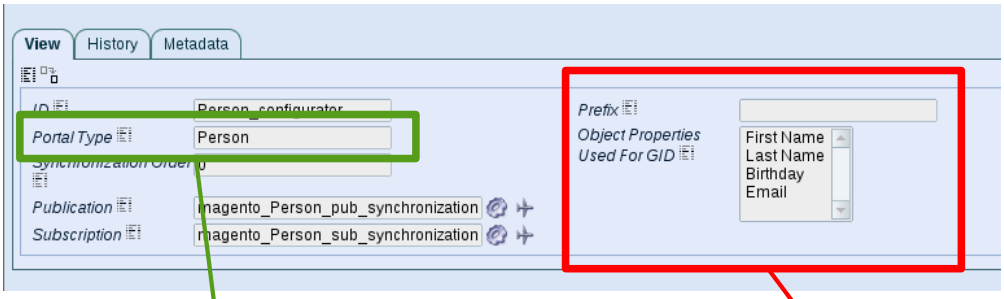
© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

The synchronization configuration tabs displays the list of configurator items that are used to defined how a the synchronization of a module will work.

Synchronization order must be defined striclty otherwise nothing will work. For example, a sale order defined a source and a destination which are persons objects. If all persons are not already synchronize, this synchronization of sale orders will fait.

Items are also link to synchronization object like publication & subscription. This link does not have to be edited.

Synchronization Configurator Item



Type of object this configuration is applied to

Define how GID is computed. Properties that can be used depends on module we synchronized

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

On the synchronization configurator item's view, everything is editable. You can define :

- ID : must be *PortalType*_configurator

- Portal Type : Type of object this configurator is related to, usually one per module. Default value must not be change

- Synchronization Order : Integer to let TioSafe the order, the lower the first.

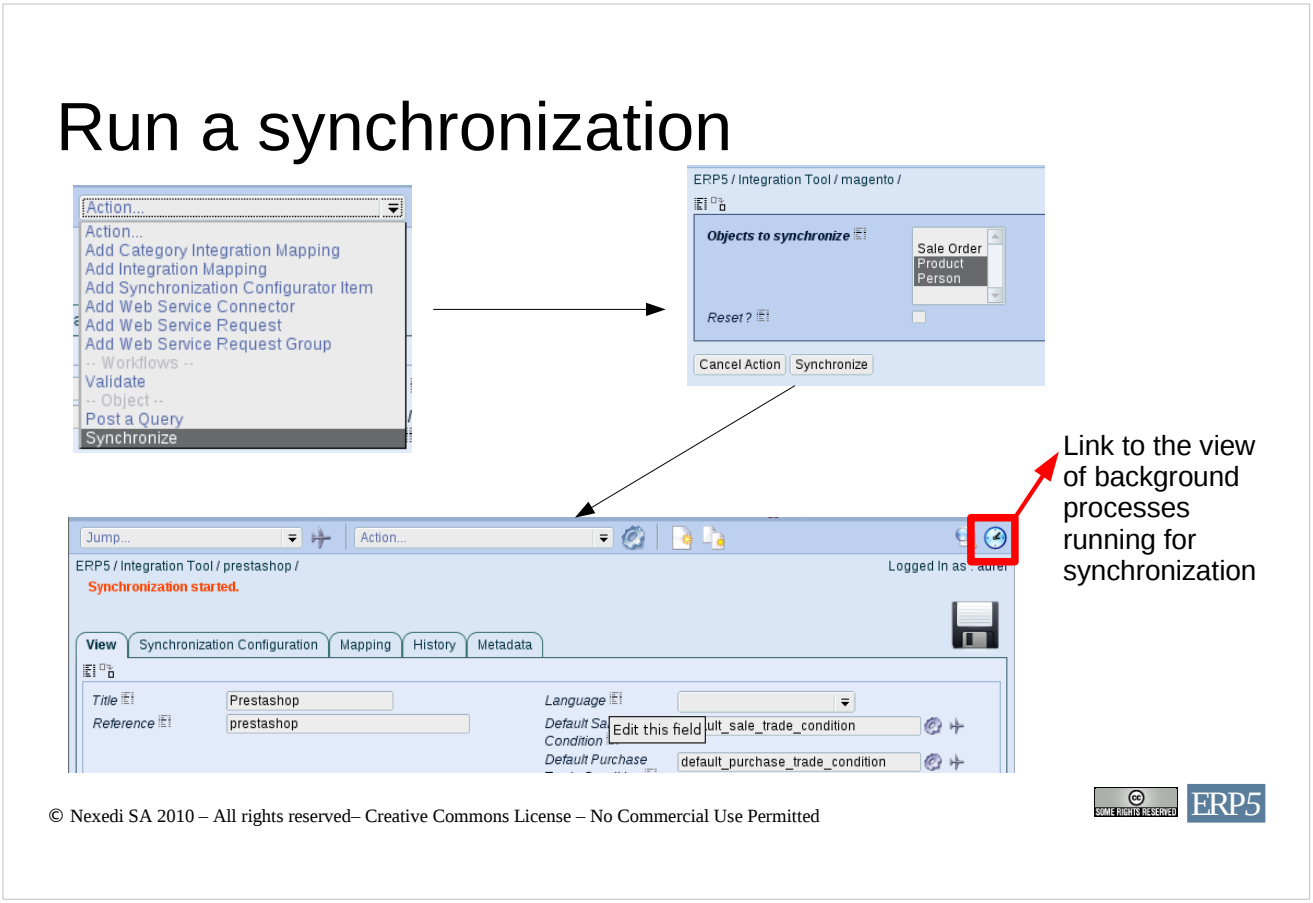
- Publication / Subscription : link to synchronization object. Default values must not change/

GID Part :

GID is used as a global ID between the SaaS application and TioSafe, this means it should depends only on properties exchanged between application so that with a given GID we can retrieve the same object the two applications. GID can be configured with this form as application constraints are not always the same. For example, on one application, all products are uniquely identified by a reference, on another, it is the ean13 which is used to do this. So

- Prefix : prefix used in GID, usefull to distinguish same kind of object coming from differents applications (sale orders coming from multiple e-commerce applicaion). This can prevent collision between reference of sale orders.

- Property : list of object properties that can be used to uniquely identify it, this can be a combination for example : first name + last name + birthdate



In order to run the synchronization process, use the Action menu ➔ Synchronize. You will be redirected to a form where you can choose the module you want to synchronize. In the end all modules have to synchronized but for debugging purpose it is possible to synchronize only one module at a time. Once running a clock is display at the top right of the synchronization site document. Clicking on it will bring you to portal_activities which basically show processes which are running in background. You will then see all activities related to the current synchronization. Once finished, this view is empty

Status of Synchronization

Overview | Activities | LoadBalancing | Advanced | Contents | View | Properties | Security | Undo | Ownership | Find | History | Workflows

CMF Activity Tool at /erp5/portal_activities

Queued Activities

Warning: Invoking or canceling activities will invoke/cancel all activities with both the same path and the same method_id.

Warning: Invoking an activity will **not** run any dependency check.

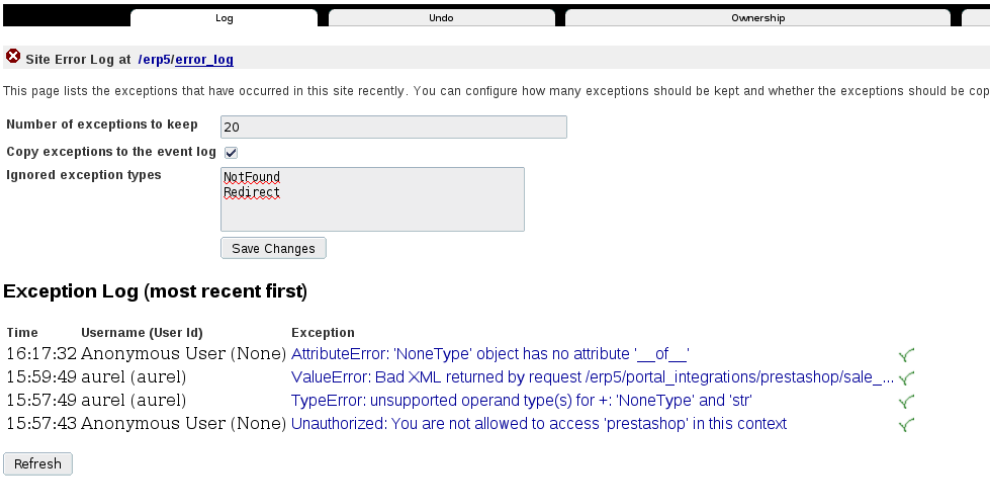
	Uid	Object	Method Id	Activity Kw	Arguments	Named Parameters	Processing Node	Retry	Processing	Call Traceback
Cancel Invoke	2214 (dict)	/erp5/portal_synchronizations	SubSync	'tag': 'portal_integrations/prestashop', 'after_tag': []	'/erp5 /portal_synchronizations /ps_person_sub'		1	0	0	
Cancel Invoke	2215 (dict)	/erp5/portal_integrations /prestashop	getTitle	'after_tag': []			0	0	0	

Refresh pages clicking “Activities” tabs until list is empty
or activity in processing_node=-2 → this means there is an error

© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted

Error view

URL :http://localhost:18080/erp5/error_log/manage_main



© Nexedi SA 2010 – All rights reserved– Creative Commons License – No Commercial Use Permitted