



D5.2.1 The Europeana Service Integration Registry

This deliverable is software.



co-funded by the European Union

The project is co-funded by the European Union, through the **eContentplus** programme

<http://ec.europa.eu/econtentplus>



Österreichische
Nationalbibliothek

EuropeanaConnect is coordinated by the Austrian National Library



ECP-2008-DILI-528001

EuropeanaConnect

The Europeana Service Integration Registry

Deliverable number/name	<i>D 5.2.1</i>
Dissemination level	<i>Public</i>
Delivery date	<i>30 May 2010</i>
Status	<i>V1.0</i>
Author(s)	<i>Theo van Veen, Thomas Beekman</i>



eContentplus

This project is funded under the eContentplus programme, a multiannual Community programme to make digital content in Europe more accessible, usable and exploitable.

Distribution

Version	Date of sending	Name	Role in project
0.1	22.04.2010	Joachim Korb	WP leader
0.1	26.04.2010	Jan Molendijk, Vassilis Tzouvaras	Technical lead, developer
0.3	27.05.2010	Jan Molendijk, Vassilis Tzouvaras	Technical lead, developer
1.0	14.06.2010	Federico Milani EC	PO

Approval

Version	Date of approval	Name	Role in project
0.3	28.05.2010	Jan Molendijk	Technical Lead
0.3	30.05.2010	Vassilis Tzouvaras	Developer
1.0	14.06.2010	Max Kaiser	PC

Revisions

Version	Status	Author	Date	Changes
0.1	Draft	Theo van Veen	22.04.2010	Initial version
0.2	Draft	Theo van Veen	20.05.2010	Comments from Jan Molendijk and Vassilis Tzouvaras
0.3	Draft	Theo van Veen	27.05.2010	Comments from Joachim Korb
1.0	Final	VPZ	31.05.2010	Layout



Management summary

The Europeana service integration exists of two parts:

- 1) the service integration software
- 2) the service integration registry

The service integration software is part of the Europeana portal. This software offers the user the possibility to use the Europeana metadata as input for external services. When and how these services are integrated is described in Service Integration Descriptions (SIDs). These SIDs follow the Schema for the Integration of Web Applications (SIWA). This schema is described in M5.2.1.¹) The portal will read the SIDs from the Europeana service integration registry or from a location as specified by the user and will allow the user to select services depending on context (search results) and conditions as mentioned in the SID. In general metadata in the full display serve as input for a service and as the context that triggers the service integration software to offer a service to the user.

The service integration software does not fully implement all SIWA features. It is advised to extend the software when more integration functionality is required.

The actual content of the registry is initially defined by Europeana. It is however recommended that Europeana will enable advanced users to share new SID's with other users in a global registry allowing all users to inspect this registry and select extra services they want to be available in the portal.

¹ This schema was developed because existing initiatives like WSDL and UDDI were not sufficient for automatic integration of services in a Web application or were not low barrier enough for this purpose.

Description of the service integration software developed for Europeana within EuropeanaConnect

The Europeana service integration consist of two parts:

- 1) the service integration software
- 2) the Europeana service integration registry with the schema from M5.2.1

The actual software consists of the following components:

- 1) A JavaScript include file (siwa.js) doing all the client side integration
- 2) The services server software that manipulate the Service Integration Descriptions (read, edit, convert, store)
- 3) Modifications in the Europeana portal to enable service integration

The basic scenario is the following (see figure 1). For each full record display the siwa.js is included and an initialisation function is called. This function will request a JSON object with service integration descriptions from the services server. The first time the service server will read the default Europeana XML service integration descriptions from a file and convert these to a JSON object.

The initialisation function scans all metadata fields in the display and checks whether they are defined in the SID as trigger. If so, a menu with links to service requests will be added to the metadata field. As soon as the user clicks such a menu option, the service request is generated based on information in the SID and the service is invoked. The types of access that will be supported in Europeana are HTTP GET, HTTP POST or JSON requests. The response is shown in a new window, new div or directly in a metadata field.

More extended SIWA features are invoking services automatically and to check for extra conditions before invoking the service or offering the link to a service to the user. It is also possible to define buttons or links to request a service on page level rather than using the menus for individual metadata fields.

The modifications in the Europeana portal needed to support SIWA are:

- 1) include siwa.js
- 2) placing metadata in spans
- 3) call the initialisation function
- 4) an optional array with extra data to be provided to the services integration software in siwa.js

Some features require spans or divs as place holder for the output of a service for example to position a thumbnail or a suggestion list. If not available the SIWA software will create one.

There will also be a facility to get an overview of all available services. Currently this can be started from within the service menu that is shown when clicking a metadata field. The user will then be able to modify the metadata fields that trigger the service, the text that is shown in the services menu and whether the service is to be invoked automatically. Additionally the user will

be able to access an external service integration registry (D5.2.2) to add services integration descriptions created by others.

An overview of the service integration infrastructure is shown in figure 1.

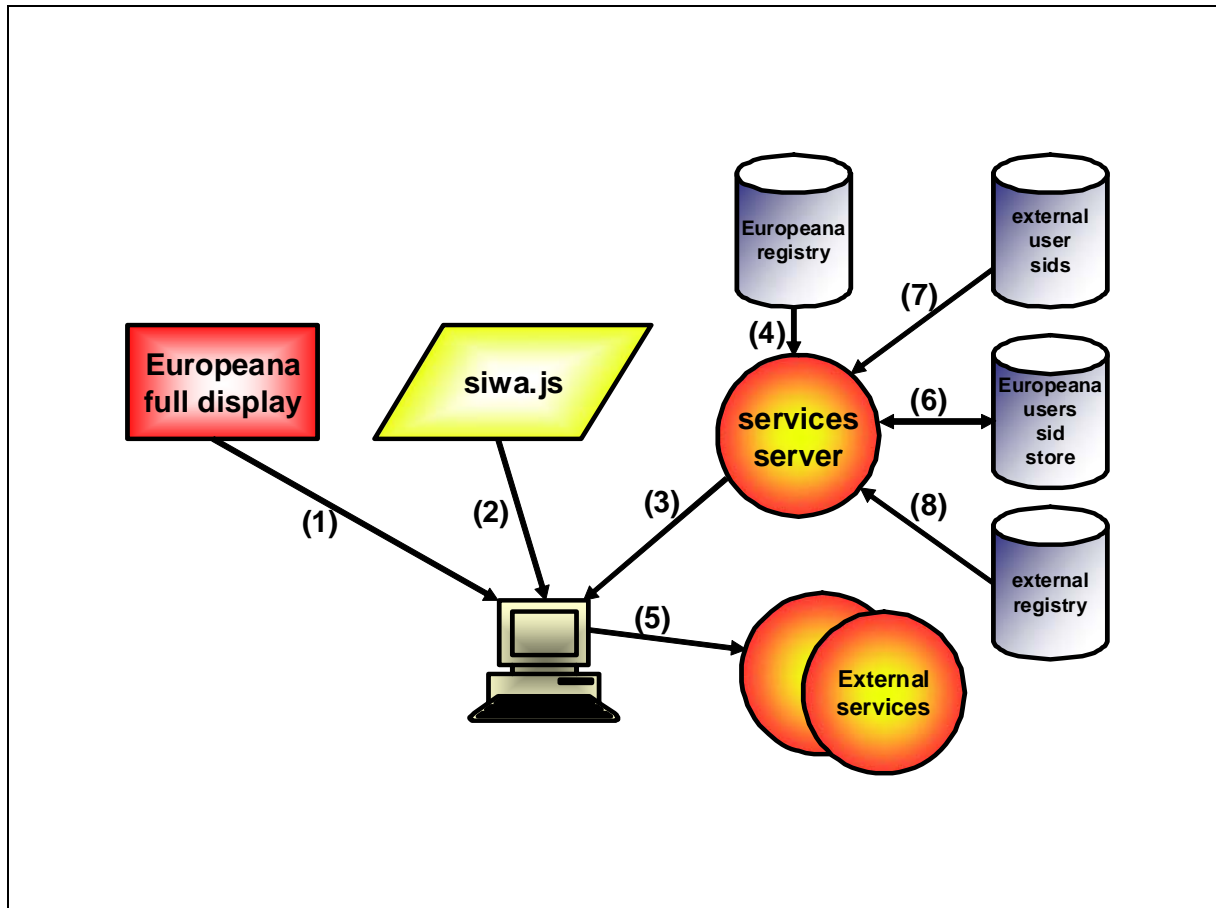


Figure 1 Overview of Europeana services infrastructure

The overall scenario for service integration based on the numbers in figure 1 is as follows:

1. A Europeana full display is requested
2. A reference to a JavaScript file is included containing all the service integration functionality
3. The service integration functionality requests the service integration descriptions from a services server that converts the SIDs to JSON format
4. Initially the SIDs come from the Europeana service integration registry
5. The service integration functionality will invoke services based on context provided by the full display and the contents of the SIDs
6. Registered users may modify SIDs and store them in a user store for usage in next sessions



7. Users may specify a private SID file that is accessible via http instead of the Europeana SIDs
8. The service integration functionality allows access via SRU to external SID registries. Users may select specific SIDs to be used during a session and registered users may store these SIDs in their personal Europeana SID store.

The Europeana registry will be a file with SIDs which are selected by Europeana and which will be available to all Europeana users. Users may propose new services but because of security risks and the risk being compromised SIDs created by users will not automatically become part of this registry. In the future Europeana might allow advanced users to add SIDs to a global registry that is accessible via SRU ²⁾ and will allow users to select their own preferred services to be integrated with Europeana. Such a registry should also be usable for other portals that support SIWA like The European Library portal.

siwa.js

Link to software	https://europeanalabs.eu/svn/contrib/kb/siwa/
Login information	EuropeanaLabs credentials
Development environment	None (Notepad++)
Programming language used	JavaScript
Application server used	None
Database requirements	None
Operating system requirements	None
Port requirements / default ports used	None
Interface	None
Licensing conditions	None

²⁾ The choice for SRU is to have a generic API rather than a proprietary API (like UDDI) for accessing searchable registries.



ServicesServer

Link to software	https://europeanalabs.eu/svn/contrib/kb/siwa/
Login information	EuropeanaLabs credentials
Development environment	IntelliJ IDEA 9.0.1
Programming language used	Java 1.6 (Spring Framework)
Application server used	Jetty
Database requirements	None
Operating system requirements	None
Port requirements / default ports used	8080
Interface	Web service
Licensing conditions	None