FUNCTIONALITY OF THE EWP NETWORK FROM A BUSINESS PERSPECTIVE

The functionality of the EWP Network is defined by its APIs (Advanced Programming Interfaces), i.e. services (specified in [GitHub](#)) which can be used by network participants to exchange data. An up-to-date list of available APIs with version numbers is available at https://developers.erasmuswithoutpaper.eu.

The EWP Network has a Development environment (DEV) and a Production Environment (PROD). New users can test their implementation at will in DEV but have to prove they have implemented the basic APIs and signed the Terms of Use agreement before they are accepted by the Steering Committee in PROD.

Services (APIs) supported by the network cover the following business areas.

1. State of the Network connection

The Registry, Discovery and ECHO APIs are the most important services of the EWP Network.

The Registry API is implemented by the Registry and is used to gather information about services delivered in the network by the hosts connected to it. The Registry offers a catalogue file with all the binding information obtained from the registered manifest files.

Each host which wants to post the supported APIs should implement the **Discovery API** (manifest file) and send the URL of this file to the Registry. Discovery manifest files inform network users about which HEIs the given host covers, which features (APIs) it implements, and which credentials clients from this host are going to use when fetching data from the EWP Network.

The EWP ECHO API requires EWP developers to design and test the authentication and security framework, which will be used for network communication. To be able to send the data via the Network, the host should implement at least basic network security protocols (more advanced ones will be needed for the exchange of data with more security demanding partners).

2. Getting acquainted with each other

Being part of the network, you want not only to share information about yourself but also get information about the other network participants. Most probably the user is a higher education institution (HEI) involved in Erasmus+ mobility. **Institutions API** and **Organizational Units API** are for sharing details on the organization, its structure (organizational units), contact data, in particular fact sheets with information directed towards incoming students.

3. Course catalogue

Higher education institutions offer courses to incoming students from the partner institutions. You may direct your students to the partner's web page or you may give them access to the course catalogue in your portal by getting courses from the partner and posting them locally. You may also

EWP 2.0. Functionality

want to get details of a particular course identified by a course code. **Course**-oriented **APIs** are used for that.

We think about extending the suit of APIs for handling courses to better suit the need of the network participants, for example to allow for remote filtering of courses.

If you are not interested in courses of your partners or do not plan to post your educational offer on the net, you may skip these APIs. This is the general rule concerning participation in the EWP Network – you do not have to implement all APIs, you only implement a subset of your choice.

4. Interinstitutional Agreements

Erasmus+ mobility is based on Interinstitutional Agreements (IIA), which are the approved (electronic) documents describing details of the planned mobilities – how many students, which study fields, for how long and when. Agreeing to and signing an IIA precedes any exchange of students/staff. The **IIA APIs** are used for sharing details among partners and getting formal approval, in particular (electronically) signed documents. Student/staff mobility is carried out in the context of a particular agreement but from the EWP point of view you don't need to share IIAs to implement and carry nominations, Learning Agreements or Transcript of Records. Most probably, however, you will need details from IIAs to control your mobilities in a local system.

5. Erasmus+ mobilities

The most important functionality of the EWP Network is to support the Erasmus+ mobility of students. The core processes of student mobility are:

a. Nominations

Students are nominated by the sending institution for outgoing mobility. The number of students and their study fields should be compliant with the cooperation conditions from the Interinstitutional Agreement. The sending institution exchanges a list of nominated students with the receiving institution to get the approval. Negotiations on the list of nominations are carried out via the EWP network using the **Outgoing Mobilities API**.

The Incoming Mobilities API is implemented by the receiving institution. It allows the sending HEI to read the data of the receiving HEI that contains information related to the outgoing mobilities of the sending institution . This may be for example the status of the nomination or arrival/departure dates.

Currently, these APIs describe mobilities of one type only – Student Mobilities for Studies. More types will be added in the future.

The nomination process creates a so-called mobility (an outgoing student with all the related attributes), identified by a unique identifier. This mobility provides a context for other processes concerning this outgoing student.

b. Learning Agreements (LAs)

A nominated student needs a detailed study programme for the agreed mobility period. A LA lists the courses which a student should take at the receiving institution. These courses, after being approved by the student, a coordinator from the sending institution and a responsible person from the receiving institution, become requirements that need to be fulfilled for the mobility to be successful.

EWP 2.0. Functionality

The Outgoing Mobility Learning Agreements API is implemented by the sending institution. It allows the receiving HEI to read and accept Learning Agreements stored on the sending HEI's servers and propose changes to them. Negotiations on the courses should be carried out via the EWP Network, leading to a list of courses approved by all three parties.

If the HEI implements both the Outgoing Mobilities API and the Outgoing Mobility Learning Agreements API, then it must ensure that every learning agreement object will have the same identifier as the corresponding outgoing mobility object served by the Outgoing Mobilities API. We say that the LA is handled in the context of the mobility.

The current version of this API does not handle signed PDFs, but that may change in the future if the need arises.

Outgoing students are recognized by the mobility identifiers but we are thinking about specifying the possibility to send LAs without the mobility context – if the need arises.

In the context of the Erasmus Dashboard, HEIs can also deal with learning agreements via the OLA tool. The Erasmus Dashboard is now being connected to the EWP network.

c. Transcript of Records

The Incoming Mobility Transcript of Records API is implemented by the receiving institution. It allows the sending institution to retrieve Transcripts of Records issued by the receiving institution for a given set of mobility identifiers.

The current version of this API allows you to send signed PDFs.

Incoming students are recognized by the mobility identifiers but we are planning to specify the possibility to send transcripts without the mobility context as well, which would cater for a much wider user scenario.

6. Erasmus+ Reporting

Erasmus+ mobilities have to be reported by HEIs to the European Commission's Mobility Tool+ platform, managed by the Directorate-General for Education and Culture Unit (DG EAC). There is a group of Mobility Tool+ APIs for getting dictionary values needed for reporting, getting data on institutions involved in the Erasmus+ exchange and the carried out projects.

The most important API, i.e. the one for reporting mobilities, is delayed but available in a draft version. It may still be subject to changes.

7. Grade conversion

Grade conversion is not only part of the Erasmus+ mobility. It is needed in all scenarios where grades issued in one grading culture are to be interpreted and converted to another grading culture. It may concern a Transcript of Records, a Diploma supplement, or any other document which carries grades.

In the current specification a Grade conversion table can be passed from the receiving institution to the sending institution together with the transcript of records.

Eventually the Grade conversion table may be added to other APIs or sent separately, on request.

8. The ESC Router and the exchange of data

The European Student Card (ESC) Router is a central system which gathers some data of students who want to obtain a European Student Card. Student data needed to issue the ESC can be sent from the home HEI to the ESC Router by means of the ESC Students API (available in a draft version).

9. Other business oriented data

The EWP Network supports the exchange of data related to Erasmus+ student mobilities, but eventually it may exchange many other kinds of student data such as Diplomas, Diploma Supplements, and Student Accommodation Data, to name but a few. Although designed to work for Erasmus+ mobilities, it will in the future be able to exchange any student or HEI-related info.

10. Change Notification Requests (CNRs)

Some APIs have related CNRs. A CNR is used for letting your partner institution know that the related objects have changed in your system and that the partner – if interested in the changes – should request the changes by calling the corresponding APIs. CNRs do not carry data, they are simple change notifications (like SMSs sent from one system to the other). In the manifest files you promise your partners to send notifications, and CNRs allow your partner to wait for a notification and then pull the data.

11. Testing tools

New developers will find support in the available testing tools.

The XML Validator is used to check the produced XML data.

The API Validator allows for the automatic testing of most of the APIs (in particular ECHO, i.e. security protocols of the network). The Validator tests whether:

- the partner's response is compliant with the XSD schema and specification (available in GitHub),
- the partner does not expose confidential data to all partners in the EWP Network,
- the partner properly implements the communication protocols of the EWP Network.

Possible test configurations:

1. The developer uses the Validator to test one's own implementation but with the use of the public installation of the EWP registry.
2. The developer uses the Validator to test one's own implementation but with the use of the local installation of the EWP registry.
   2.1. The developer uses the Validator run from a console.
   2.2. The developer uses the online Validator (run from the browser).
3. The EWP Network administrator uses the Validator to test a new partner institution before accepting it for in the production environment.

12. Extending the EWP Network

The EWP Network **is open** in the sense that new services (APIs) can be added freely, when needed. The sky is the limit. There is therefore no reason to build another network for the exchange of

academic data. The EWP Network can consolidate services offered/used by higher education institutions into one digital platform for academia.

Extra info:

Janina Mincer-Daszkiewicz, University of Warsaw – EWP technical development leader
jmd@mimuw.edu.pl