

Open APIs in Information Systems for Higher Education

Ricardo Barata, Sérgio Silva, David Martinho, Luís Cruz, Luís Guerra e Silva

Instituto Superior Técnico, Av. Rovisco Pais, 1049-001 Lisboa, Portugal

Keywords

Open, API, OAuth2, information systems, innovation, community.

1. ABSTRACT

For many years now information systems have played a central role in the daily operation of any higher education institution. Most institutions already have a wide range of applications at their disposal, from Learning Management to Course Management systems as well as other Administrative systems. In some cases these systems are proprietary applications, and users can only use them “as is”. Even in cases where these systems are open source, it is difficult, if not impossible, for the users to contribute back to the system with anything other than a bug report or a feature request.

Consequently these systems evolve very slowly, and their development is typically dictated by the priorities of each school, identified by their respective boards. This results in hindered system innovation and a waste of potential resources, especially in schools that have engineering or computer science courses. In recent years the massification of handheld devices (such as smartphones and tablets) coupled with the availability of relatively easy to use software development kits widened this gap of wasted resources.

In December 2013 our school, Instituto Superior Técnico (IST), released the first version of an open API for its academic information system, which is an instance of the open-source project FenixEdu^[1]. With this API anyone can now develop their own applications using the data stored in the academic information system, without having to be limited to the particular technology adopted for its implementation. This API also allows for easier integration with other systems. We hope this open API will promote creativity while harnessing the potential development capabilities of the entire community, from students, to teachers, to third party software vendors.

The API is coupled with the OAuth2^[2] protocol and each endpoint has appropriate access control, such that only applications that have been granted access to a user's data can access it. In the upcoming months we will be making more endpoints available to increase the coverage of the API and will also be hosting events to promote its usage.

By the end of the academic year we hope to have concrete examples of how the community chose to use the new API as well as preliminary data to evaluate its impact on the school. Our objective is to foster the growth of an ecosystem of applications based on the API that complement or complete the schools information systems.

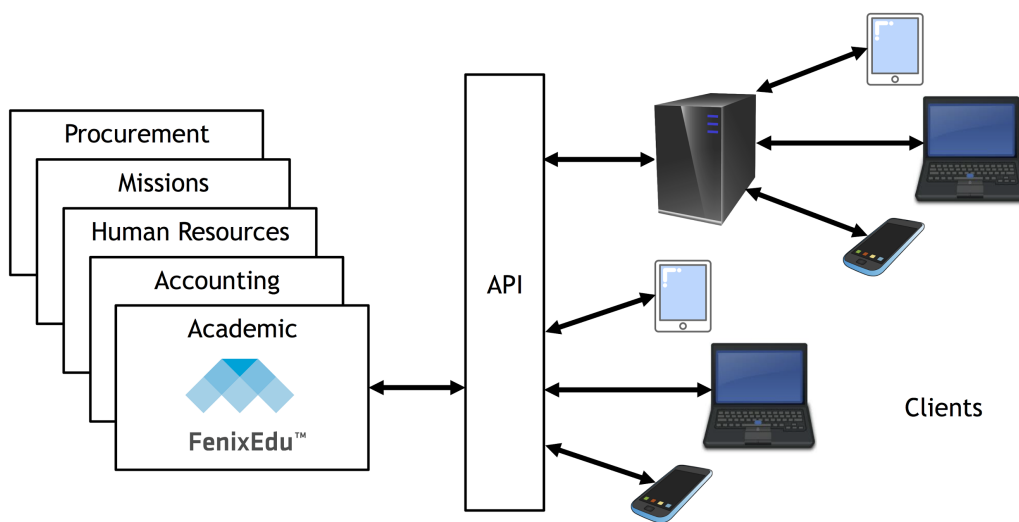


Figure 1. Illustration of API usage in IST.

2. BACKGROUND

The landscape of information systems in higher education is very diverse. To the best of our knowledge, a unified solution, capable of handling all the necessities of such institutions, is yet to be developed. Therefore, most institutions have a few distinct information systems with different levels of integration and interoperability. While some institutions have fully integrated platforms others choose to use independent software modules to solve different problems. In all these cases, general common components such as Content Management Systems (CMS), Learning Management Systems (LMS) or Enterprise Resource Planning (ERP) modules can be identified. Some institutions choose to develop these modules in-house, others choose to outsource them, and others opt for a community based development approach.

Regardless of the adopted approach, all these applications have a development community that is limited in one form or another. Proprietary applications have a development community limited to the company that developed the software and that extends at most to its partners. Community or federation based development is in some cases limited to the respective members. Even in open source projects, the development community is limited to the developers with knowhow of the underlying technology, and contributions are typically subject to an acceptance process.

Consequently these systems evolve at a fairly slow pace. In most cases new features are a consequence of administrative decisions and alternate solutions are rarely developed and explored. End users have little bearing in regards to development decisions and are stuck with whatever software has been developed or adopted.

At IST, we have a mix of in-house open source developments and contracted proprietary applications. The schools administrative board dictates development priorities. In both cases the response time to development requests fall short of users expectations. When features are developed, these don't always meet the needs of every user, or if they are optimized for one user, they may be troublesome for another.

The recent expanse of smart devices and massification of tablets and smartphones is another area where our systems have fallen short of expectations. Furthermore this is a market where user creativity can lead to innovative new solutions.

The FenixEdu project develops open source applications for higher education institutions. It's flagship application is a bundle that includes an Academic Management System, a CMS and an LMS, that have all been customized to IST's requirements and integrated with other information systems used by the school.

3. CHANGING THE PARADIGM

Information systems are, at their essence, vehicles for storing, processing and communicating information and knowledge. Modern information systems must be available to users every day, 24 hours a day, and users are no longer content with only accessing systems via a personal computer, they must be available to a user on whatever equipment is most convenient. The interfaces provided to the user may either simplify his life or seriously hinder his productivity.

To promote innovation, and simplify the incorporation of new and alternate solutions to the school's information systems, IST decided to "open" its academic information system. In December 2013 the school published an API for its academic information system, allowing any developer to register their applications and access the API. Any IST user can now use any registered application as long as they grant that application access to the subset of information requested by that application relative to that user.

Since IST is a school of engineering, where most users have significant knowledge on information systems, we foresee the development of a multitude of client applications, not only by commercial entities but also by students and faculty. The availability of a wide range of Software Development Kits (SDKs), for different technologies and platforms, that can make use of such an API enables drastic reductions in development times and allows any *tech-savvy* person to easily create their own applications.

4. API IMPLEMENTATION

When publishing an API for integration with external applications, one should follow industry standards. Therefore, for the API we adopted a REST *web service* based solution, resorting to OAuth2 for handling authentication and authorization.

The major challenges when implementing the API are twofold: choosing what endpoints should be provided and defining the structure of the returned information (both syntactically and semantically). An important aspect that must be considered is that implementation changes should not disrupt the API, since that can potentially break client applications. Therefore, extra care should be taken when architecting the API so as to allow the implementation to evolve, without breaking the API's "contract" with existing clients. Any changes to the API, other than the addition of new endpoints, or new fields to existing endpoints, should constitute a new version of the API that is published and maintained in parallel with previous versions. Both versions should be made available as long as there is any client that still uses the previous version (backward compatibility).

The first version of the FenixEdu API provides the following endpoints:

GET /about	GET /person
GET /academicterms	GET /person/calendar/classes
GET /courses/{id}	GET /person/calendar/evaluations
GET /courses/{id}/evaluations	GET /person/courses
GET /courses/{id}/groups	GET /person/curriculum

GET /courses/{id}/schedule	GET /person/evaluations
GET /courses/{id}/students	PUT /person/evaluations/{id}
GET /degrees	GET /person/payments
GET /degrees/{id}	GET /spaces
GET /degrees/{id}/courses	GET /spaces/{id}
	GET /spaces/{id}/blueprint

The full specification is available on the following site: <http://fenixedu.org/dev/api/>. An example request to the API would be the following HTTP GET request:

<https://fenix.tecnico.ulisboa.pt/api/fenix/v1/degrees?academicTerm=2013/2014>

which would result in the following output displayed in Figure 2:

```
[ {
  "id": "2761663977313",
  "name": "Lean Six Sigma Black Belt",
  "acronym": "CELSSBB",
  "academicTerms": [ "2013/2014", "2012/2013" ],
  "academicTerm": "2013/2014",
  "type": "BOLONHA_SPECIALIZATION_DEGREE",
  "typeName": "Curso de Especialização",
  "url": "https://fenix.tecnico.ulisboa.pt/cursos/celssbb",
  "campus": [ { "type": "CAMPUS", "id": "2465311230081", "name": "Alameda" } ],
  "info": {
    "description": "As pessoas entram e saem das organizações...",
    "objectives": "É objetivo do curso munir o aluno com capacidades...",
    "designFor": "Sendo de carácter geral, o curso tem o potencial...",
    "requisites": "Podem candidatar-se a este curso os licenciados...",
    "professionalExits": "",
    "history": "",
    "operationRegime": "Início: 10.01.2014 • Fim: 27.09.2014 ...",
    "gratuity": "O valor do curso é de 4.000 € ...",
    "links": "http://www.isixsigma.com/"
  },
  "teachers": [
    { "name": "Luís Manuel Mendonça Alves", "username": "ist13976", "mails": [ ], "urls": [
  ]}
],
  ...
]
```

Figure 2. Example output of an invocation to the FenixEdu API

5.1. Client Registration and Authorization

Any developers of third-party application that need to access the API must first register the application as a client of the API. To do so, developers must be registered with the academic information system

(FenixEdu), and select the “Personal > Manage Application” menu entry in their home page. Here users can register their applications by providing basic information, as illustrated in Figure 3.

Personal Space

Create Application

Logo: No file chosen

Application Name:

Description:

Site URL:

scopeList:

- All
- schedule - Information about your academic schedule and calendar.
- payments - Information about payments and debts.
- info - Your **public** personal information: name, emails, ist-id, photo and web page.
- evaluations - Information about your tests and exams.
- curricular - Curricular information: enrolled/teaching courses.

Redirect URL:

Figure 3. Example form for registering API clients.

Application registration requires the developer to provide basic information that is necessary to identify the application including a logo, an application name and a description. Additionally, a site URL can be provided so users can learn more about the application before choosing to use it. The developer will also be required to provide the list of data scopes that the application needs to access. Each scope corresponds to a subset of the user’s private data. Finally a redirect URL indicates where users will be redirected when they grant the client application access to their information.

This information is used by the FenixEdu API to identify each third-party application and allows users to grant access to their private data through the OAuth2 authorization protocol. Each registered application will be provided a client identification code (key) that it must use in all its interactions with the API.

When using a third-party application for the first time, a user will be prompted to authorize it to access his private data. During this prompt the user is informed of the scopes that the application is requesting access to. If the user does not authorize the application, it will not be able to retrieve any data that pertains to the user. If the user grants the authorization, the application will be able to retrieve data limited to the scopes with which the application was registered. Figure 4 illustrates the dialog that is presented to the user to authorize third-party applications to access his private data.

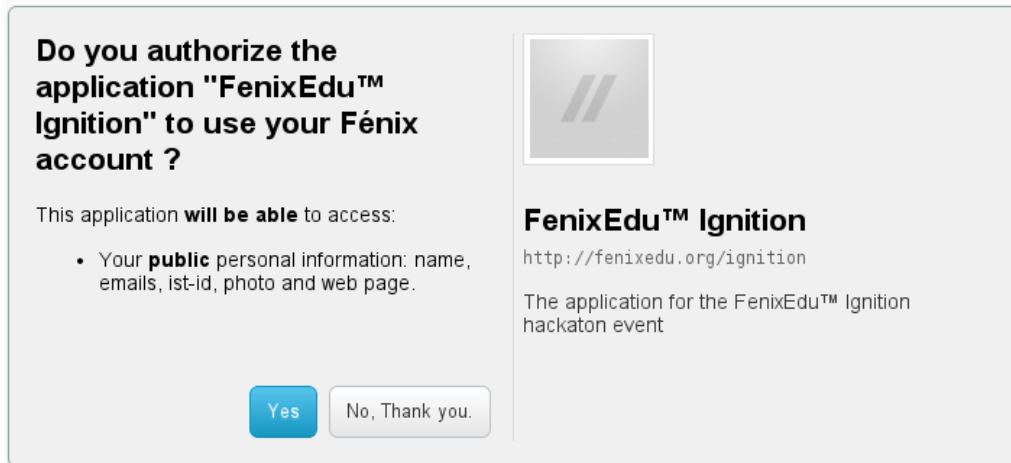


Figure 4. Example of the dialog for authorizing client applications to access the private data of a user.

In this example, “FenixEdu Ignition” is the name of the application client of the API. This application will only be allowed to access the public information of the user (name, e-mails, id, photo and web page URL).

6. Reaching Out to the Community

To help kickstart the development of clients for the FenixEdu API, we have published a few SDKs that provide a basic platform for interacting with the API. Currently we have published SDKs for Java, PHP, Python and Lisp.

With the availability of the API to the entire academic community, we hope that our users will take initiative and develop their own ideas of how to use or integrate solutions with the school’s academic information system. To further attract the interest of our students (in particular of Computer Science students), we hosted a contest in February of 2014, where students were invited to submit their ideas and prototypes of API clients. While the awards will be mostly symbolic, we hope the initiative will stimulate the creativity of the academic community.

Since the API was published in December 2013, we have had over 220 developer registrations to use the API. While many applications have been registered so far, 10 of these applications are already being used by people other than their creators. Figure 5 displays a chart of the number of invocations to the API per day since its launch. On May 2014, there are 70 registered applications using the API. The most used application is a mobile application developed by one of our computer engineering

students which is being used by 805 active users.

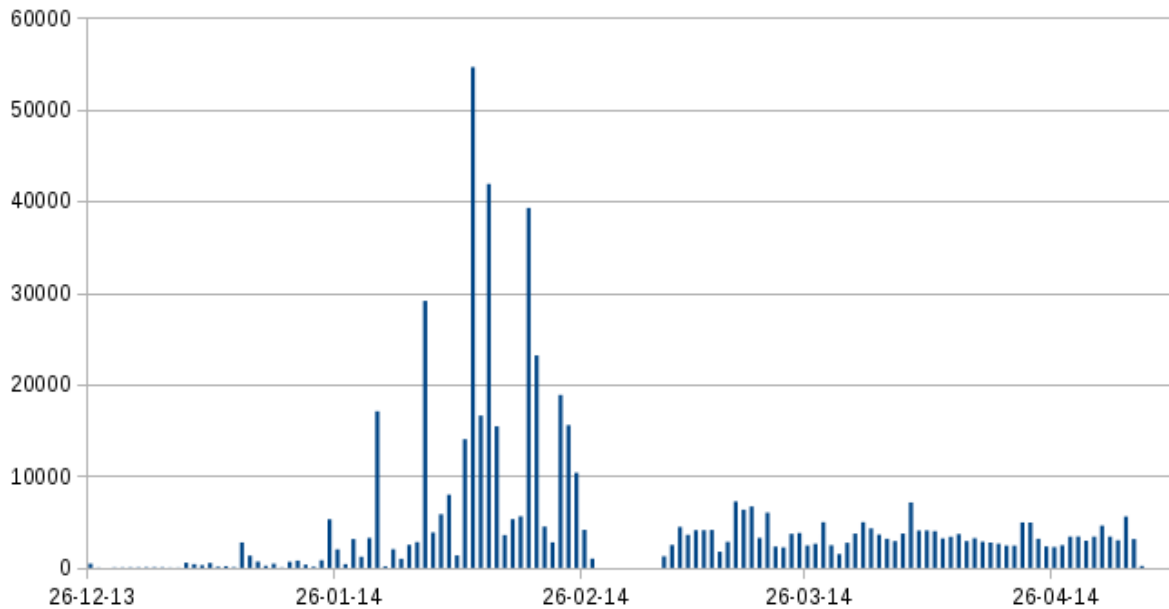


Figure 5. Number of requests to the FenixEdu REST API per day since December 2013 until May 2014.

7. Measuring Results

Determining the success or failure of this project may not be straightforward. While we hope to see the emergence of a few widely used client applications within the upcoming months, the true success of the project will be dictated by the usefulness of those applications to the academic community. Will these client applications of the FenixEdu API improve the lives of the school's community?

A related metric, that will be much easier to assess, can be obtained by gauging the usage of the API, both by the number of clients, number of end users and respective activity. We intend to make this information publicly available on the website of the project in the upcoming semester.

8. Conclusions

Opening the information systems of higher education institutions through an API on which others can build applications that complement or augment the initial system promotes creativity and initiative in a community oriented environment. This strategy results in a larger development community in which relations between developers and users are strengthened. Users will have more choice in regards to

what applications they use. This may potentially change the way the community interacts and will allow good ideas to be implemented and provided to the community at no additional cost to the institution.

Implementing an API for existing information systems is relatively easy and can be done at little cost, regardless of the technology underlying the existing information systems. The availability of such an API has the potential to change the landscape of the school's information systems, resulting in an information rich ecosystem.

References

[1] FenixEdu Open Source Project: <http://www.fenixedu.org>

[2] OAuth2 protocol: <http://oauth.net/2/>