KU LEUVEN

FACULTY OF SCIENCE

Cultivating APIs in the cultural heritage sector

Lessons from an internship at Europeana

By Jolan Wuyts

Blauwe Vogel, 15e E BCE. Glasdia van een Minoïsch Fresco, oorspronkelijk gevonden in Knossos. Aanwezig in de Collectie van Kunstgeschiedenis KU Leuven, 1839-1939. Fotograaf onbekend. Aangeleverd aan Europeana via het Photoconsortium.

Internship Coordinator: prof. Danny De Schreye
Internship mentor: Nienke van Schaverbeke

Thesis based on an internship presented in fulfillment of the requirements for the degree of Master of Science in the Digital Humanities.

Word of advice: This thesis is best enjoyed in digital format. It deals with User Experience and web design topics using existing online examples, which are best viewed alongside the text.

Keywords: API, APIs, Cultural Heritage, GLAM, digital humanities, art history, digital history, libraries, digital library, digital archive, archives, digital museum, museums, digital gallery, galleries, interface design

Biases: Europe-centric, Europeana-centric, focus on The Netherlands in specific and Western Europe in general,

Abstract

The outcome of this research thesis is a set of suggestions for Cultural Heritage Institutions on how they may improve the impact of their API(s). A study of the field of Cultural Heritage APIs, and hands-on work with one CHI's API documentation and presentation were used to come up with these suggestions. The result of this foray into cultural heritage APIs is the insight that cultural heritage APIs are very different from other, mostly commercial, APIs and should therefore be built, presented, and documented differently.

An initial literature study documented previous research dealing with cultural heritage APIs. Their conclusions and arguments were discussed critically and taken into account when looking at contemporary use cases. The API or other open data documentation of several CHIs were studied to get insight into which kind of users CHIs cater their open data to, what the opportunities are for CHIs in the presentation of their open data, and how this presentation could be improved.

The Europeana APIs were taken as a case study. The documentation and presentation of the Europeana APIs were compared against that of other Cultural Heritage APIs, and a number of possible shortcomings in the API documentation, presentation and UX design of the Europeana APIs were identified.

Abbreviations used:

API: Application Programming Interface
CHI: Cultural Heritage Institution
DPLA: Digital Public Library of America
GLAM: Galleries, Libraries, Archives, Museums
LAM : Libraries, Archives, Museums
LOD : Linked Open Data
SPARQL: a recursive acronym for "SPARQL protocol and RDF Query Language"
URI: Uniform Resource Identifier
UX: User Experience

Introduction and objectives

This Master thesis is a written account of my work and research as an Intern at Europeana. I did this internship as the final leg of my Digital Humanities study programme at KU Leuven, for which I moved from Belgium to The Hague. My internship consisted of a three-month fulltime employment at Europeana. From September until December 2017 I worked in the Europeana Collections team, coached by my internship mentor Nienke van Schaverbeke. My tasks were very diverse, which makes it hard to write a coherent research paper that describes everything I did during those three months. Instead of doing that, I decided to focus on a relatively small part of my work at Europeana and expand it into a broader research paper. **My other tasks are explained very briefly at the end of this thesis in annex 3. They include subtasks, skills I used from my DH study programme, and skills I learned while doing those tasks.**

The meat of this thesis focuses on APIs, which I hope a lot of other people find as exciting as I do. Work on the APIs was mostly executed under the instruction of Hugo Manguinhas, the product owner of the Europeana APIs. My work with Hugo to improve the Europeana API documentation started with me wanting to use the Europeana Search API to make a simple call. I quickly realised that even I, who was surrounded by specialists on Europeana, and arguably knew a bit more about using API's than the average user of the platform, found it extremely hard to figure out how to use Europeana's Search API. Feedback from other colleagues indicated that in their opinion the API documentation was hard to read and not up to date, and when talking to Hugo Manguinhas he voiced the same concerns. With these concerns coming from inside the organisation, I conducted a relatively compact study of the field to get a grip on how other cultural institutions were presenting their APIs. The outcome of this literature study can be found in the first part of this thesis' Results section. After familiarising myself with the field of cultural heritage APIs, Hugo and I sat together to create some operational tasks to improve the Europeana API documentation and presentation. The insights from this work on the Europeana APIs is described in the second part of this thesis' Results. A more in-depth look into how this literature study and editorial work was conducted can be found in the following section, Methodology.

Aside from the operational tasks that we set out to complete, I formulated a few research questions as well to try and answer along the way. These were:

What are the target users of APIs for cultural heritage?
What are the possibilities that cultural heritage APIs give to those target users?
How can we present cultural heritage APIs in a way that attracts these target users?

These research questions focus both on users and on CHIs, who connect and interact through the CHIs online platforms and services. This paper will invariably deal with user experience and web design topics because of the nature of the research questions. More specifically, there will be a focus on user groups and how to cater to those user groups, and on how CHIs can make conscious decision about how they design their web platforms to provide a better, more impactful user experience.

Methods and Context

Context: Cultural Heritage APIs

What is an API?

API stands for Application Programming Interface. It's a set of rules and standards that makes it easier for computer software programs to communicate with each other. There are a lot of different kinds and flavours of API out there, but here I want to focus on what most cultural heritage institutions use when it comes to APIs: **RESTful web APIs** that give people access to the metadata related to digitized cultural heritage objects available on the web. A REST API is a kind of API that connects over the HTTP protocol to handle requests and return information. The world wide web uses REST protocols: You give them a request over HTTP, e.g.. 'http://www.google.com', and the response is Google's homepage. Cultural Heritage APIs also handle requests, and mostly respond with metadata about cultural heritage objects, often in XML format.[1] Metadata can be defined as 'data about data'. If the data is an image, its metadata might include: an extension, like .JPG, indicating it is an image file; a timestamp of when the image was created, modified, uploaded, or changed; the size of the image file, e.g. 40 kilobytes, etc.

Europeana has one REST API, that breaks down into several smaller APIs that serve different purposes. With Europeana's APIs you can search through all of Europeana's metadata collection, find all the metadata from a single record, annotate existing records with your own notes, and even download all of Europeana's data if you wanted to! The Europeana APIs are part of Europeana as a platform. Multiple services are built on top of this platform. The Europeana Collections, www.europeana.eu , is the most well-known service.

Methodology

The research conducted to come to the results of this thesis happened in three stages: a literature study, a study of the field, and assessment and editing of the Europeana API documentation.

The literature study surveyed different papers, blogs, and posts by people working in the cultural heritage sector on the topic of APIs and data accessibility. The goal was to gain insight into how CHIs made choices about their digital cultural heritage. I tried to learn why CHIs would choose to develop an API, or why they wouldn't, and what influences those choices. The literature studied covers different geographical regions (e.g. the United States, Europe, and Australia), is recent but still captures different moments in the development of digital cultural heritage in the past ten years, and was published on a mix of platforms (academic journals, blogging platforms, CHI websites…).

A study of the field of CHI websites today was conducted to identify best practices of API documentation development, and to find the similarities and differences in the presentation of CHI APIs across different platforms. A wide range of CHI websites were inspected: The National Library of Scotland, the Internet Archive metadata API, the Harvard Art Museums API, Canadiana, the Tate Collection, Trove, DigitalNZ, Finna, Rijksmuseum, and Open Cultuur Data. Every CHI platform was graded on a checklist of items, with every item being checked if the CHI provided it. A few examples would be: does the CHI provide downloadable datasets? Do they have an online console for their API? Do they show examples of reuse of their data through the API?. Based on this checklist, a list of suggestions was made of items that CHIs regularly didn't include in their API or other data accessibility documentation.

[1] 'Representational State Transfer' on Wikipedia, last accessed on 6/12/2017, https://en.wikipedia.org/wiki/Representational_state_transfer

Lastly, the Europeana APIs were studied as a use case. The Europeana APIs went through the same checklist that the other APIs went through in the second methodology step. With the migration of the Europeana API documentation to a new website (pro.europeana.eu), I had the opportunity to help implement some changes in the existing documentation to try and improve the presentation and user experience of the webpages. The new webpages were designed to visually show examples of reuse, included an API console, aggregated information and examples in dropdown menus as to not overload the user, included a link to libraries and plugins, and a few other changes. A showcase webpage was designed to give an idea of how the current documentation pages could be improved further.

Results

Context: other studies of Cultural Heritage APIs

> "Since open data is, in theory, accessible to anyone with Internet access, cultural institutions have to consider how to format, organize, and develop their open data platforms to reach their intended audiences—whether it's the noobs, the tech savvy, or anyone in between."[2]

Andrea Ledesma wrote a Medium blogpost in March of 2016 on the journey that the Harvard Art Museums went through in the development of their own API.[3] They asked themselves the same main questions that are asked in this thesis: *"What types of projects can take advantage of open data access to museum collections? Who is interested in accessing open data for museum collections, and how does "tech-pertise" factor in? What are the risks and benefits to consider when deciding to make an institution's data available through an open data platform?"* These questions seem to fall into two broad categories: one about the users of a cultural heritage API, and another about the openness of data. Ledesma doesn't give an answer to who she thinks the actual users of the Harvard Museums API are, but she does make an important point about what kind of users they would have to be.

> "while there's a definite learning curve for working with APIs—especially for those in the humanities without a programming background—we found that the sheer amount of information that can be pulled through these systems is enough motivation to learn these skills."[4]

Ledesma and her colleagues, who were humanist researchers with only limited technical expertise, undertook the journey of trying to extract meaningful data out of their own API to gain insight into their collection. **They experienced first-hand that there is, indeed, a learning curve for using APIs. It was the value of the results they gathered that validated all the effort they had put into learning to use the API in the first place.** Once they had taken this first step in understanding and using their own API, I'm sure it was much easier for them to conduct similar research about their own collections. **The motivation for an average user to start learning how to use an API is probably quite low if they don't have a clear view of what the possible outcome might be.** The whole endeavour might seem pointless, without any merit. It is here that the CHI has the opportunity to show a user what the possibilities of learning to use their API are. **By showing example applications, the motivation to take that first step might be given to more users visiting the CHI's API documentation. A great first move here is to conduct internal experiments** using the API, just like Ledesma did with the Harvard Art Museum API. These experiments can then be used as a first example of how the API can be used, and what results might be procured. These kinds of examples give users that already have a high level of tech-pertise a first insight into how the data is structured, and it sets users with a low level of tech-pertise on their way to trying out the API for the first time. Coupled with easy to follow tutorials to recreate these examples, users of all technical levels will be able to interact with the CHI's API. The need for tutorials will be expanded upon later.

The second category of questions and concerns Ledesma voiced had to do with the openness of data. Ledesma rightly points out that the more complex a CHI's data is, the more complex its API will have to be, resulting in a higher learning curve for those wanting to use the API. Ledesma identifies

[2] Ledesma, Andrea. "How We Learned to Stop Worrying and Love Open Data: A Case Study in the Harvard Art Museums' API." Https://medium.com/@andrea_ledesma. Andrea Ledesma on Medium, March 2016. https://medium.com/@andrea_ledesma/how-we-learned-to-stop-worrying-and-love-open-data-a-case-study-in-the-harvard-art-museums-api-893c3f40ecb7
[3] Ibid.
[4] Ibid.

two different reactions to this reality. One, exemplified by the remarks by Hogarty from the SFMOMA, suggests that simplifying the data for use through an API will make it easier to use for the users. While this may be true, Ledesma points out that this sacrifices a lot of the richness and openness of the data. The transparency of a CHI's data is lost when it is altered and/or simplified before users can access it, which defeats a big part of the purpose of creating an API in the first place. Furthermore, this discourages internal use of the API, since it is an inferior entry point into the data compared to other entry points that might be available internally. Not using one's own API internally also goes past an exciting opportunity for CHIs: creating a single standardised point of entry into one's platform, accessible both internally and externally. I would argue there is no better way to show to users that a CHI is transparent and public-facing than giving users the exact same data accessibility and tools as the staff of the CHI itself. This opens the way for outside projects to be incorporated into the CHI's services directly, and mutual enrichment and learning opportunities. The other reaction came from Cooper Hewitt from the Smithsonian Design museum, who came with a plan to foster public interaction with an API for a wide range of users. The plan consists of making databases more user-friendly in general, creating easy to follow guidelines, and adding amalgamated fields, aka buckets, to create more flexibility in API querying. It is especially this last bullet point that goes a long way to solve the issue of making complex data easy to use. **Creating combined fields makes querying easier and more intuitive, without sacrificing the option for finer granularity in query calls.**

> "tradeoffs exist between the openness of the data and the openness to the public, and institutions are finding a balance that fits their missions."[5]

Of course, not every cultural heritage institution wants their data to be completely open and accessible. They might have copyright restrictions on their metadata, or might want to keep control over their own content. This is a different, albeit equally important, discussion to have. We won't discuss this much further here, and instead assume that CHIs that have developed or are developing an API want their content to be as open and restriction-free as possible. The case of openness will be discussed in the context of Europeana in the third part of this Results section.

Another great example of staff creating their own example experiments on their APIs is Tim Sherratt from Trove. The 'historian and hacker' Sherratt was Manager at Trove from 2013 to 2015, and created several applications using Trove's API during that time. In a talk on March 1, 2013, he said:

> "Trove has an API which makes doing this kind of stuff easier, but when I started doing this Trove didn't have an API. This meant I had to resort to all sorts of horrible things like screen scraping, following HTTP headers, etc. […] In doing this stuff of reverse engineering, basically hacking, I don't think I'm doing anything wrong, I don't think I'm breaking any laws or breaching any terms of service, although it's hard to be certain. I'm very public about what I do, I share my products and code, I talk about what I do, but I still feel sort of guilty with doing this kind of stuff, like I'm sneaking around. […] It makes you feel like doing anything other than using the supplied user interfaces is bad. People ask me what the copyright status is of the objects in my applications, and my answer is 'I don't know, I'm not certain.'."[6]

At this point, Sherratt had just been or would shortly be hired by the National Library of Australia to manage Trove, its digital platform.[7] Before that, he had already created an unofficial API for Trove, and had created applications using Trove's data by screenscraping and hacking his way into their content. How Sherratt used Trove before getting officially working there is, in itself, an argument for

[5] Ibid.
[6] Tim Sherratt, 'Exploring with Confidence', 1/03/2013, http://discontents.com.au/exploring-with-confidence/ , last visited on 06/01/2018.
[7] Tim Sherratt, '8 months on', 23/12/2013, http://discontents.com.au/8-months-on/ , last visited on 06/01/2018.

CHIs to open up their data and provide access through an API. In Ledesma's rundown of potential risks in using an API, she mentioned 'loss of control' as one of the most interesting ones. After looking at how Sherratt was able to gain access to and control Trove's data by reverse engineering their interfaces**, I would argue that APIs provide the opportunity for CHIs to regain control of their (meta)data.** Sherratt's hacked together applications are an example of how only providing data through a certain interface doesn't necessarily mean that a CHI has total control over that data. On top of that, Sherratt had no idea what the copyright status was of the objects he scraped from the Trove website. An API extends the opportunity to clearly state the copyright status of the metadata it provides, and the data associated with it.

When Tim started working for Trove, his arsenal of applications and experiments expanded. Following Europeana's example he created a Trove API console to create easily shareable examples of API calls and responses.[8] **An API console allows users to experiment with calls and queries immediately, without having to program their own wrapper for the API. It also makes sharing responses, showing examples, and explaining fields and functions much easier**, as Sherratt mentioned. Apart from the API console Sherratt also created or contributed to other experimental applications with exquisite names such as the Serendip-o-matic, the Trove Zone explorer, and the Trove News Bot.[9] Sherratt's numerous experiments reveal his excitement of working with Trove as a platform.

Sherratt put it simply and effectively with the **phrase 'Portals are for visiting, Platforms are for building on.'** It might also have been Dan Cohen who coined the phrase, since Sherratt mentions it during his 2013 presentation while talking about Dan Cohen's explanation of the DPLA API.[10] Where Europeana.eu, aka the Europeana Collections, is a portal to its data and metadata, Europeana offers more than just its portal. Europeana, DPLA, Trove and any other CHI that offers an API are effectively platforms. They provide the possibility of building new applications, portals, services, connections on top of their API(s). **The importance of being a platform for reuse of CHI data lies for a great part in allowing anyone to create something new with that data.** Other people might think of doing something with cultural data that cultural heritage institutions might never think of by themselves. Another important point to take into account, Sherratt also notes, **is the ability to connect APIs together**. APIs make it possible to link platforms together, to connect the data of different providers and aggregators into applications that search through data across borders, organisations, continents.[11]

In 'Reprogramming the Museum', Luke Dearnley from the Powerhouse Museum in Australia recounts the process the museum went through to end up with their API[12]. With the goal of giving others access to their data, the museum decided to create **a data dump of all their metadata. It soon became clear that this had its drawbacks,** which will be described in the next section of this thesis. Dearnley describes the advantages the Powerhouse Museum saw in providing users with an API instead of a single data file:

> "The apps people built using our dataset would have benefitted from an API. They needed to ingest our entire dataset to build their app regardless of whether they were interested in every field or even every record. Using the API they could have pulled in merely what they needed. Furthermore the data in these apps gradually becomes out of date and, once

[8] Ibid.
[9] Ibid.
[10] Tim, Sherratt. "From portals to platforms: Building new frameworks for user engagement." Hamilton, New Zealand, 2013. https://www.nla.gov.au/our-publications/staff-papers/from-portal-to-platform
[11] Ibid.
[12] "Reprogramming The Museum | Museumsandtheweb.Com." Accessed December 20, 2017. https://www.museumsandtheweb.com/mw2011/papers/reprogramming_the_museum.

> again, if using the API they would be drawing on the most up-to-date version of our collection data instead."
>
> One of the main advantages the API has over the data dump is the ability to track use. [...] Given that we had little idea how much the API would be used initially and into the future it was specifically designed to operate on technologies which would allow us to easily relocate it from local servers to cloud-based hosting. So one big reason to track the API's usage trends is to help determine on-going hosting requirements. Tracking is also necessary for providing evidence of usage to support the business case and funding of these projects."[13]

Following the logic that CHIs want to grant access to their collections for reuse and research, tracking that reuse and research of cultural heritage is an important metric for the success of an API. It's not enough to keep track of how many users request an API key, or how many API calls are made. Most of those keys will be used once, or not at all, after their request. Similarly, only a small amount of API calls will actually be used for reuse and research of cultural heritage. From August to October 2017, 113 new Europeana API keys were requested. During that same time, the total amount of API keys that made more than 5 API calls a day on average actually decreased, from 79 to 76. Requesting API keys or registering minimal API calls doesn't mean that new apps or tools were or will be developed using the CHI's platform.[14]

It's important to know when new apps, services, papers, or other products are released that use your data. Being on the lookout for new things that incorporate your API is one thing, but an easier and more rewarding strategy might be to encourage your API users to engage with the CHI, send feedback, and let the CHI know when they made something new with the CHI's data. This community involvement may be as simple as creating a discussion forum for users of an API, or as extensive as organising hackathons to let users get hands-on experience with using your data. David Weinberger posited that 'A library platform should be measured less on the circulation of its works than in the circulation of the ideas and passions these works spark'.[15] **A crucial impact factor of a CHI's collection is exactly this spark of ideas and passions in others. It seems like a hard metric to quantify, but keeping track of what others do with your platform and encouraging users to enrich the CHI's data with their own contributions.** Sherratt describes the emergence of #TroveTuesdays, a movement initiated completely by Trove users without any connection to the National Library of Australia. Another Trove example is Ravelry, a community of knitters that explores Trove's historical newspapers and magazines to find knitting patterns.[16] As mentioned before, showcasing examples of what other applications have done with a CHIs API is a good way to inspire and generate insight for other users and potential users of an API. On top of that it also shows off the impact a CHIs data has, how it has inspired other creators to make something new and to add value back to the CHI. It's a unique opportunity for a CHI to demonstrate its function, its use, as a 'possibility engine' (a term also used by Tim Sherratt in his 2013 presentation) by showing what it has made possible.

Dearnley goes on to describe the choice the Powerhouse museum now faced: continue serving their data over the API as well as offering a downloadable dataset, or ceasing to provide the dataset and only allowing access through the API. The argument that digital humanists preferred a readily available dataset for data analysis won them over to keep on providing both forms of access. Having to write code to access the museum's data was a huge barrier to access, writes Dearnley.[17] Is this

[13] Ibid.
[14] Internal Europeana data, last recorded on January 5 2018.
[15] http://lj.libraryjournal.com/2012/09/future-of-libraries/by-david-weinberger/#_
[16] Tim, Sherratt. "From portals to platforms: Building new frameworks for user engagement." Hamilton, New Zealand, 2013. https://www.nla.gov.au/our-publications/staff-papers/from-portal-to-platform
[17] "Reprogramming The Museum | Museumsandtheweb.Com." Accessed December 20, 2017. https://www.museumsandtheweb.com/mw2011/papers/reprogramming_the_museum

really the case though? Yes, having to write code to access data is a barrier to access, but there are other ways of lowering that barrier, the easiest of which for the CHI as well as the user is providing an API console. **The more important barrier to access with an API compared to a downloadable dataset lies in limiting the amount of API calls that can be made from a single API key, and the length of an API response.** A digital humanist might even prefer using an API to create a dataset for analysis, since the API output can be formatted and edited to fulfil the needs and goals of the researcher. It might result in less need for data cleaning by the digital humanist. A researcher might only want to look at a small part of a CHIs collection, which means they would have to first extract that small subset of objects from a downloadable data file, and reformat and clean it before use. With an API, the researcher could create a call that only extracts the data one is interested in and immediately format the response of that call in a preferred feature set and file extension, including relevant metadata fields and leaving out the irrelevant ones. APIs are perfect for this kind of small-scale digital humanities research**. I would, however, also argue that continuing to provide data in downloadable datasets is a very sensible endeavour, if the CHI has the possibility to do so.** Depending on how big a CHI's data stores are, limits on API calls or the maximum number of rows a response provides can make it impossible to perform big data analysis. Trying to get a copy of a CHIs complete dataset might prove impossible through an API, or could at least take an immensely long time. Lifting the limits on the maximum amount of API calls or keys, or allowing infinitely large responses, doesn't seem like a good idea either. This could open up the API and the database behind it to malicious attacks, or could throttle access speeds to the database.
Not every CHI can simply give access to their data in a single downloadable file. When there is a lot of data still in copyright, a CHI might still consider providing only their metadata as a file. If the metadata isn't in Open access either, or is used for commercial gain, data files might just not be feasible. In these cases, controlling access to data through limits on API usage is a great compromise that benefits both user and cultural institution. **A last solution to Dearnley and co.'s issue is releasing curated datasets specifically catered towards researchers.** As described above, researchers might be interested in a single topic, artist, object type… Creating curated datasets and releasing those as downloadable files, with clear rights statements attached, are a great way to cater to the research community. They're also another perfect way to showcase the interesting and/or remarkable content in the collections of a CHI. Europeana research regularly releases new datasets on a certain topic, referring to the portal to explore these small collections of objects. Attaching a ready-made CSV file with the metadata for all of these objects would be an interesting next step in this service.

As the importance of digital collections and the access to those collections has become increasingly clear for cultural heritage institutions, it might be said the need to advocate the creation and maintenance of cultural heritage APIs isn't needed any more. Indeed, there are a slew of cultural heritage APIs available today. Mia Ridge, Digital Curator at the British Library, keeps an updated list of almost every possible GLAM API in existence. This crowdsourced collection can be visited and interacted with on her Wiki page, 'museums and the machine-processable web'. Long not all cultural heritage institutions have an API. Not a single cultural heritage institution in Belgium possesses one, for instance. On top of that, in Mia Ridge's list there are a lot of cultural heritage institutions that have started to open up their collections digitally but still have a long way to go before they'll be at their desired level of openness and accessibility. And that is, ultimately, the goal: not forcing every cultural heritage institution to create their own data model and whip up an API, but to help CHIs become as open and accessible as they want to be.

On top of all the opportunities and possibilities cultural heritage APIs offer that have been stated above, two more arguments might drive home the relevance of APIs for culture in 2018 and beyond. Firstly, machine learning techniques are increasingly being experimented with in the field of cultural heritage. Two exciting recent examples of Neural Network research being applied to cultural heritage

objects are the INCEPTION project and the SIAMESE project.[18] Both used Convolutional Neural Networks (CNN) cultural heritage objects, albeit for different purposes. INCEPTION's goal is to realise a way to create time-dynamic 3D models of cultural heritage artefacts and environments. They trained a Neural Network to recognise different parts of cultural heritage architecture, for instance to detect if a picture contains an image of a stained glass window in a church. This is a first step towards automatic detection and (re-)modelling of physical cultural heritage. The SIAMESE project applies a Convolutional Neural Network to images of historical newspaper advertisements. The Network identifies the shapes and objects in a certain newspaper advertisement and then suggests other advertisements that contain similar objects. Someone looking at an advertisement of a certain brand of car might be interested in other advertisements, perhaps from other time periods or newspapers, that also contain that brand of car. The outputs of both of these projects are no more than experiments and proof of concepts, but they're the first stage of a wealth of innovation coming to digital cultural heritage. Machine Learning techniques could completely change the way anyone finds, interacts with, researches, and reuses digital cultural heritage. With most Machine Learning techniques, the model first needs to be trained on a set of objects before it can actually predict new data. This is supervised learning, and for it to work, researchers need to have access to big datasets of labelled data, i.e. data that has already been described by humans. CHIs often already hold troves of these kinds of data, but don't allow access to them in the way that machine learning researchers need. **Opening up cultural heritage data through APIs provides these researchers with datasets that they can use to continue working on combining Machine Learning/Artifical Intelligence with Cultural Heritage data.[19]**

> "A linked dataset converts a basic catalog of cultural heritage items into RDF triples using a predefined vocabulary. These datasets can then be matched with other RDF triples to offer a richer cultural heritage experience. Linked open data releases each museum, archive, or library's collection from its silo, allowing it to be placed in the context of similar cultural heritage collections and knowledge sets from around the world."[20]

The importance of linked open data, or LOD for short, is especially apparent in cultural heritage institutions. CHIs often harbour objects that relate to objects residing in other cultural institutions, for one. Secondly, a lot of CHIs use different standards and models for describing and preserving their objects. Publishing digitised cultural heritage objects in RDF triples, as described above, allows for standardisation and interoperability between CHI data. It also connects objects digitally that might be very disconnected physically: A Van Gogh in the Metropolitan Museum of New York might be far apart from another painting by Van Gogh in Amsterdam, but in Linked Open Data Space both works will point to the same URI of their creator. CHIs are increasingly publishing and maintaining their digital

[18] Wuyts, Jolan. "Using Computer Vision Tools for Historical Newspaper Analysis: SIAMESE and Europeana Newspapers." pro.europeana.eu, December 1, 2017. https://pro.europeana.eu/post/using-computer-vision-tools-for-historical-newspaper-analysis-siamese-and-europeana-newspapers, last visited on January 8 2018.
Llamas, Jose, Pedro M. Lerones, Eduardo Zalama, and Jaime Gómez-García-Bermejo. "Applying Deep Learning Techniques to Cultural Heritage Images Within the INCEPTION Project," 25–32. Lecture Notes in Computer Science. Springer, Cham, 2016. https://doi.org/10.1007/978-3-319-48974-2_4.

[19] Other exciting research initiatives that use machine learning in some way are worth mentioning:
The Replica Project, that uses machine learning to show differences and similarities in shapes of visual artworks: https://seguinbe.github.io/2016/12/19/replica-project-status-and-roadmap.html
Oceanic Exchanges is another project that looks at historical newspaper data to study 'the global connectedness of 19th Century newspapers": http://oceanicexchanges.org/
Yasser, A., Kathy Clawson, Chris Bowerman, and M. Lévêque. "Saving Cultural Heritage with Digital Make-Believe: Machine Learning and Digital Techniques to the Rescue." Sunderland University, St Peter's Campus, 2017. http://hci2017.bcs.org/.

[20] Marden, Julia, Carolyn Li-Madeo, Noreen Whysel, and Jeffrey Edelstein. "Linked Open Data for Cultural Heritage: Evolution of an Information Technology," 107–112. ACM, 2013.

cultural heritage in Linked Open Data formats, although this is quite a new development in the field of digital cultural heritage. A lot of institutions are in different stages of opening up, publishing, granting access to their collections as Linked Open Data. The will to provide access to collections as Linked Open Data often brings with it the need for an API to access that data. The first stage of Edelstein et al.'s six stages of Linked Open Data for Cultural Heritage states:

> "The first stage to a successful linked open data project is releasing a dataset that is usable for other projects. Ideally datasets are converted into RDF triples and shared via an open API or SPARQL query endpoint."[21]

At this point, quite a lot of Cultural Heritage institutions have made their data available as Linked Open Data, which has led to a solid foundation of knowledge in connected web graphs.[22] The semantic web for cultural heritage isn't an idea any more, it's a reality. Querying this knowledge graph, as well as connecting to other knowledge bases happens through APIs. If a CHI has gotten to this stage of providing their data as Linked Open Data, connected to the other huge knowledge bases of the semantic web, accessible through an API, this means a user now has all the tools to build innovative new things with what the CHI has provided.[23]

.

[21] Markhoff, Béatrice, Thanh Binh Nguyen, and Cheikh Niang. "When It Comes to Querying Semantic Cultural Heritage Data," 384–94. Communications in Computer and Information Science. Springer, Cham, 2017. https://doi.org/10.1007/978-3-319-67162-8_38.
[22] Ibid.
[23] Two great papers that have both been cited in the above text are a good starting point for more information on the semantic web for cultural heritage:
Marden, Julia, Carolyn Li-Madeo, Noreen Whysel, and Jeffrey Edelstein. "Linked Open Data for Cultural Heritage: Evolution of an Information Technology," 107–112. ACM, 2013.
Markhoff, Béatrice, Thanh Binh Nguyen, and Cheikh Niang. "When It Comes to Querying Semantic Cultural Heritage Data," 384–94. Communications in Computer and Information Science. Springer, Cham, 2017. https://doi.org/10.1007/978-3-319-67162-8_38.

Study of the field: other Cultural Heritage datasets and APIs

Just datasets

A great deal of the more public-facing cultural heritage institutions provide access to their data and metadata in some way. Mia Ridge's list is a testament to that. Providing access doesn't always happen through an API. The National Library of Scotland, for instance, has taken to publishing some of their data as packaged datasets. By the end of 2017 they planned on publishing 13 datasets with Open Data, following their own Open Data Publication Plan. The subject of these datasets ranges from information about the National Library itself (like the list of purchases by the library over 25.000 EUR since 2016) to arrays of metadata on historical out-of-copyright data (like this great dataset on a digitised collection of Soviet posters from 191-1930). Making your data accessible through open datasets is a great start, but has quite some drawbacks. To list a few of them:

- The CHI's data will have to be hosted by anyone who wants to re-use this data in an online project.
- Zipped dataset files are static and cumbersome to update
- Curated datasets only represent a tiny part of all the open data a CHI potentially has
- Dataset files on a single subject are created for very specific use cases. There are not a lot of ways to reuse this data outside of its intended use.

These issues don't occur when a CHI uses an API to grant accessibility to their collections. Data is hosted by the CHI themselves, and this data changes when the CHI updates their database. This way, the data accessed through the API is always up-to-date, and can open up a much larger part or even all of the CHI's available data. The possible use cases of data extracted through an API are also much broader. Since the users are the ones who construct the API calls, they are effectively creating their own custom up-to-date datasets every time they make a call. The Powerhouse Museum went through the same thought process, and ended up creating an API for their collections.[24] As the Powerhouse museum also realised, having a downloadable dataset is better than having nothing. On top of that, it is still often worth it to keep providing a downloadable dataset even if you have an API that gives access to the same data. Datasets have their own merits and use cases, and shouldn't be seen as inferior to providing an API. Other notable organisations providing datasets for their collections are Open Cultuur Data[25], British Library Data[26], the Central Library of the National Hungarian Museum[27], and The Tate Collection[28].

The Digital New Zealand (DigitalNZ) API

There is a lot to say about the architecture of an API in and of itself, but this thesis wants to put the focus not on how an API is developed before being launched, but how it is presented to users after launch. Having the data and the platform is one thing, making that platform as easy and inviting to users as possible is another. Most cultural heritage APIs are similar in architecture: they are REST APIs that require an API key with minimal restrictions that fetches data via GET calls. Often there are a few different kinds of calls that can be made. A simple or advanced search will search all the records for a query and possible other parameters you set. A record lookup might return all the metadata for a single record. Finally, a third option might be to download predefined datasets by calling on the ID number of that dataset. These three options are exactly the ones that the DigitalNZ API provides, the API for the Digital New Zealand Project. Digital New Zealand's API is currently in its third version, having been around 10 years now. In 2008, the Digital New Zealand project first

[24] "Reprogramming The Museum | Museumsandtheweb.Com." Accessed December 20, 2017. https://www.museumsandtheweb.com/mw2011/papers/reprogramming_the_museum
[25] Open Cultuur Data Netherlands (http://www.opencultuurdata.nl/) and Open Cultuur Data Belgium (https://www.opencultuurdata.be/)
[26] https://data.bl.uk/
[27] https://mnm.hu/en/museum/library/open-bibliographical-data
[28] https://github.com/tategallery/collection

launched the API, becoming one of the first APIs to collect a nation's worth of digital content.[29] In 2017, Digital New Zealand refreshed their website to become more user-friendly and more generally pretty. The API and its presentation also got an overhaul. This new design of the DigitalNZ API documentation is the first use case we'll be discussing.

First of all, the DigitalNZ API pages are accessible in one click from the homepage (Figure 2.1.). [30] It isn't tucked away in a submenu, but features prominently in the main navigation bar. This says a lot about how DigitalNZ might think about their API. The navigation bar item that links to the API pages is called 'Developers'. This is clearly an important user group they're catering to with their API. This doesn't mean only professional developers should click this link, but it does imply that this part of the website is intended for people with a certain level of technical expertise. Another section lower down on the homepage also promotes the API, but in a different way. Under the fold, the title 'Create Something with us' lists a few options on how to get involved in DigitalNZ (Figure 2.2.). The third one of those option is 'Use our data. Use the free digitalNZ API to make creative websites, mash-ups, apps and more. You can see we've got some great API user examples.' This wording seems like it wants to appeal to creative users[31]. In conclusion**, DigitalNZ's homepage has two direct entries to their API pages: one in the navigation bar, starting a user journey for developers and other users with high technical expertise, and another one below the fold starting a user journey for creative users.**

When talking about user groups, it's important to reflect on how describing users might create a simplified and biased version of reality. When talking about 'marketing to creative users as well as developers', this might imply that both those user profiles can't exist in the same person. It's important to keep in mind that users aren't one-dimensional. A creative user might just as well be a developer who likes to use cultural heritage APIs after work hours. Someone who works in web development might have never touched an API before and therefore have very low technical knowledge when it comes to working with APIs. **Two important factors are at play here: the motivation of the user, and the technical expertise of the user.** A developer's motivation to visit an APIs documentation page might be to see what has changed in the newest version of the API, because the developer is in charge of maintaining an application that uses that API. A creative user's motivation might be to explore what others have made with the API and get inspired to create a website of their own using the same techniques. Similarly, a user with low technical expertise will more likely benefit by being presented with clear steps on how to use the API (first request an API key, then start reading the documentation, use the code examples to create your first call on the API console...) than a user that has a lot of experience with APIs. A user with high technical expertise will probably find more use in a clear and concise documentation page that lists all the parameters, responses, error codes… In the end, it's important to remember that user groups and users aren't a correct representation of the people visiting your website, and that people might fall into one, more or none of the user groups you're marketing to. This doesn't undermine the fact that it's useful to think of user groups to have an idea of what kind of people you want to attract. For the purpose of presenting APIs, this thesis will focus on the motivation of users, and their level of technical expertise.

[29] DigitalNZ, "Our History", on https://digitalnz.org/about/our-history , last accessed on January 10 2018.
[30] https://digitalnz.org
[31] Future Physical defines the creative user in their user analysis report as 'anyone whose participation in an interactive artwork, installation or performance involves a level of interaction that results in them creating any form of content which did not previously exist without their input.' After adapting this definition to include more than physical art, and simplifying it for brevity, a working definition could be: a creative user is someone who adds value to a platform or process by creating new content through participation.
Future Physical "The creative user: a broad definition", Future Physical User Analysis, http://www.futurephysical.org/creativeuser/downloads/pdfs/33.pdf , last accessed on January 11 2018.

Once inside the DigitalNZ API pages, there really isn't that much content (Figure 2.3.). The webpages are very simple, minimal, and pretty to look at. The landing page for the DigitalNZ API describes a few things right off the bat: What the DigitalNZ data contains, what you can do with the API, a call to action to use the API to create something new, links to examples, version info, and the links to the first steps a user should take to start using the API. The landing page gives all of this information in very compact and simple wording. Here, again, the page seems built to cater to both developers and creative users. The API terms of use and version history are important for contractors or developers maintaining older software using the API. The call to action, link to examples, and links to first steps, are a good way to motivate creative users to get started, or to guide users with low technical expertise.

DigitalNZ provides a list of examples of applications that have been built using their API (Figure 2.4.). The list of examples isn't very long, and there isn't a lot of visually attractive content on the page. Two videos do show different and exciting applications of the API. Pond is an education resource that allows you to search through DigitalNZ to find relevant digital content.[32] Lifelines is a physical table in the National Library of New Zealand that 'uses the DigitalNZ API to connect visitors to their heritage through dates, places and people'.[33] The innovative use of the API in these resources make them interesting picks to feature on an examples page like this one. Furthermore, if users are inspired to create something after looking at these examples, the code samples and libraries page provides a few different code samples and even a tutorial to create a simple search widget using the DigitalNZ API. A simple search widget might not be what the user wanted to create, but it's a great starting point for people with low to medium technical knowledge who can then build off of these sample applications.

There are more options available for users who don't really know where to start if they want to use the DigitalNZ API. The 'Getting Started' page provides step-by-step instructions on how to get an API key, what to consider when thinking about using DigitalNZ's data, and how to start using the API (Figure 2.5.) They also urge users to get back in contact with DigitalNZ and let them know what the user is working on. **The importance of closing this feedback loop as a metric of impact of a cultural heritage API was mentioned before. Including a call to action to get back in touch and tell the CHI wat you're working on as a user with the promise of promoting innovative re-use is a great way to close this feedback loop.**

Lastly, there's the actual documentation pages (Figure 2.6.).[34] There are three short webpages for the DigitalNZ API, corresponding to the three kinds of calls you can perform: Search for a query, Search for a single record, or return a set. The pages are very simple, concise, but still hold all the information you would need to interact with the API. It's easy to see which parameters are required and which are optional. Example parameters and calls are included to clarify and illustrate. The response elements and format are clearly outlined.

The strength of DigitalNZ's API documentation and presentation lies in its simplicity and clarity. The documentation is easy to find, makes sure it caters to users with low as well as high technical knowledge, and is easy to understand. It's a pity that there doesn't seem to be an online console to test queries on, or an FAQ for more advanced issues a user might encounter. Users with high technical knowledge might expect more detail from the documentation, like error response codes, API call limitations, etc. Other CHIs from differing backgrounds were studied for this field work, but DigitalNZ's API documentation was described in-depth because it was recently renewed and is a

[32] Pond Overview, https://www.n4l.co.nz/pond/ , last accessed on January 11 2018.
[33] DigitalNZ, "API examples", https://digitalnz.org/developers/api-examples-in-use , last accessed on January 11, 2018. More information can be found in National Library of New Zealand, "the Lifelines Table", http://natlib.govt.nz/visiting/wellington/the-lifelines-table , last accessed on January 11, 2018.
[34] DigitalNZ "API docs V3", https://digitalnz.org/developers/api-docs-v3 , last accessed on January 11, 2018.

good example of best practices, while still having a lot of opportunities for improvement. Other CHIs with interesting API platforms that were studied are the DPLA, Trove, and Finna.

Case Study: Europeana

In September of 2017 Europeana moved part of its services to pro.europeana.eu. Different websites used to cater to different user groups, before integrating everything except Europeana Collections into pro.europeana.eu. Where originally the Europeana API documentation could be found at labs.europeana.eu, it is now filed under 'Resources' on pro.europeana.eu. This move was done to increase the visibility for and traffic to the API pages. Labs.europeana.eu was specifically marketed to creative users and developers. Other user groups, like researchers, were directed to research.europeana.eu or another satellite website. With the integration of all these different facets of Europeana into Europeana Pro, it became possible to present the API to more than just developers and creative users.

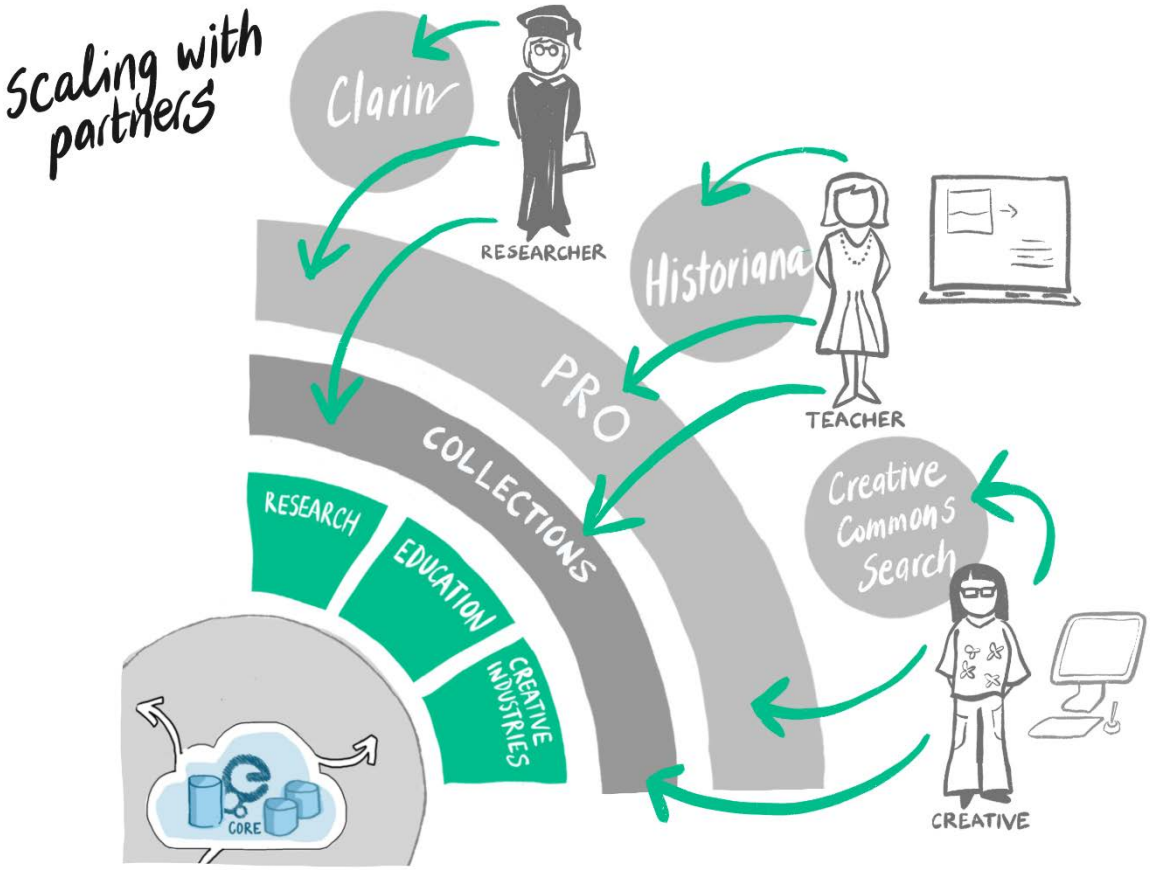


Fig 1.2.[35]

Europeana Pro is supposed to appeal to researchers, people in education, and creative users, as well as developers (Figure 1.2.). The Europeana Pro website was designed for these different kinds of users, who will take different paths through the webpages dependent on their motivations and interests. A creative user might, from the homepage, go to 'Creative Industries', and find a link to the API through there. A teacher might go to the education landing page of Pro, and find the API under the resources listed there as well. In the end, even though the API webpages are more clicks away from the homepage than was the case on labs.europeana.eu, there now are more paths, each geared towards different kinds of users, that lead to the same API webpages. This allows for a more in-depth

[35] Visual by Elco van Staveneren for the Europeana Strategy 2020, www.denkschets.nl , CC-BY-SA

look at how users behave on the webpage, where they might be coming from, and ultimately what they might be interested in. Since the new website was launched it has become clear that from the users that entered the Europeana APIs landing page, most of them (36,48%) had clicked through from the creative industries landing page.[36] Since the new website has only been launched for a few months, it might be too early to conclude from this data that most of the users that visit the API webpages are users from the creative users group. The creative industries landing page is also where users who still surf to labs.europeana.eu get redirected to, so the more plausible explanation would be that most of the current visitors of the Europeana API webpages are users that had visited the API pages before.

From the inception of the Europeana APIs, building the APIs to be usable by a broad range of people had been the goal. In "A portrait of Europeana as an Application Program Interface", a paper published in 2010 by several staff members from Europeana at the time, a few use cases for the Europeana API were described.[37] The use cases included an external application for Moodle, an educational platform. Another use case was catered towards humanities computing. The possibilities for researchers, educators, and creative users have been kept in mind during the building of the API, so it would be only natural to also keep them in mind in the design of the API documentation pages. The different parts of the Europeana Pro website already deliver services relating to the API that are specifically geared towards specific user groups: the Europeana Research landing page points towards curated datasets that are perfect for analysis and research by digital humanists, and the Europeana Labs environment is filled with examples of new applications and projects that use the Europeana API to creatively reuse Europeana data.

The data and metadata available through the Europeana APIs are complex. Europeana follows the Europeana Data Model, a European standard model for cultural heritage data. It aims to be an exhaustive solution to the issues brought up by the vast range of metadata standards used in European Cultural Heritage Institutions.[38] The final data and metadata stored in Europeana and accessible through Europeana's APIs is a combination of existing metadata standards and vocabulary such as SKOS (Simple Knowledge Organisation System)[39] and Dublin Core[40], and the Europeana, or EDM metadata. On top of this data complexity, there are also a wide a range of different calls that can be made with the Europeana API. Europeana has Search, Record, and Sets call options, just like DigitalNZ. But Europeana's data is also searchable by Entity, is completely downloadable via the OAI-PMH service, is available in RDF format through its SPARQL API, and supports user-generated content with the Annotations API. The prime issue with Europeana's API documentation is presenting it in an understandable and easy-to-use manner, while at the same time capturing and explaining its full complexity. The API documentation should ideally be understandable and usable by people with low technical knowledge of APIs as well as people with high technical knowledge, and it should inspire and motivate people with different motivations and backgrounds to use Europeana's APIs. Due to the complexity of the data and the API architecture, users with high technical skill will expect very thorough and extensive documentation pages that explain the API architecture in as much detail as possible. On the other hand, a user that wants to try and build his first search widget with the Europeana API before going on to something more complex, will benefit by having an easy step-by-step guide that doesn't describe all the complex parameters and call possibilities.

[36] Google Analytics data, from the data of the new website launch (September 11 2017) until January 11 2018.
[37] Concordia, Cesare, Stefan Gradmann, and Sjoerd Siebinga. "Not Just Another Portal, Not Just Another Digital Library: A Portrait of Europeana as an Application Program Interface." *IFLA Journal* 36, no. 1 (March 1, 2010): 61–69. https://doi.org/10.1177/0340035209360764
[38] More information can be found in the Europeana Data Model documentation: "EDM Documentation" on Europeana Pro, https://pro.europeana.eu/resources/standardization-tools/edm-documentation , last accessed on January 11 2018.
[39] SKOS reference, https://www.w3.org/TR/skos-reference/ , last accessed on January 11 2018.
[40] Dublin Core metadata initiative, http://dublincore.org/ , last accessed on January 11 2018.

First off, the APIs have an introduction page (figure 3.1.).[41] It includes a call to action, a prominent link to get an API key, a link to the terms of use, and a rundown of the different APIs Europeana has. This page is kept short and visually attractive as to not overload the user with information immediately. For people with low technical expertise, another page serves as an introduction to the Europeana APIs and as documentation on the Europeana Data Model (Figure 3.2.).[42] A downside of combining these two functions in a single page is that this can be quite intimidating to users with little to no knowledge of the Europeana Data Model or of APIs. The documentation still lacks a user-friendly tutorial that goes through the steps of getting an API key, choosing which API to use, and making a first call. These tutorials have the ability of lowering the barrier to access significantly.

Once a user has chosen which API call method to use, they can navigate to the documentation page for that method (Figure 3.3.). The pages clearly state what the function of each call is, and which API call best to use to best meet the intentions of the user. For every call, a few innovative examples of applications using that specific call are shown to serve as inspiration to the user. In the actual documentation information, a lot of effort has gone into describing the details and complexity of the API call as exhaustively as possible, to cater to developers with high technical knowledge. The downside of this is that the documentation pages become extremely long and demanding to process. There are examples for most of the parameters, and there is a console to test calls on, but these become inconsequential in the deluge of parameter information. The combination of catering to users with low technical knowledge as well as users with high technical knowledge in the same web page is unattainable, since these different users have very different needs.

A mockup of a webpage was designed to show how a documentation page would look like if it was specifically catered towards users with lower technical expertise than the users that the current documentation pages cater to (Figure 1.3.). The showcase page was designed as the documentation page for Europeana's Linked Open Data API endpoint, or SPARQL endpoint.[43] The page begins with a call to action sentence to motivate users to explore the API documentation. The introduction explains what the SPARQL API does, and points the user to other API methods to make sure users are choosing the right API method for their purpose. Three visually presented options are supposed to be starting points for users with different technical levels of expertise. People with no experience whatsoever would have the option of going through an interactive tutorial where they learn the basics of API usage, and make their first API call using the API console. This would require no coding, and no downloads. After this step, users could join people with intermediate technical expertise in the second step, where they explore the most useful parameters and how to use them. These parameters would have clear examples explaining them, and an embedded API console would allow testing with those parameters. An FAQ would try to solve problems users inevitably would run into. A third option would contain all the more detailed and complex facets of the API architecture. Dropdown menus would be used to keep the webpage clutter-free and show the user only what they need to see, removing all unimportant information.

[41] Europeana, "APIs", https://pro.europeana.eu/resources/apis, last accessed on January 11 2018.
[42] Europeana, "Introduction to Europeana APIs", last accessed on January 11 2018.
[43] The current SPARQL API documentation: Europeana, "SPARQL", https://pro.europeana.eu/resources/apis/sparql , last accessed on January 11 2018.

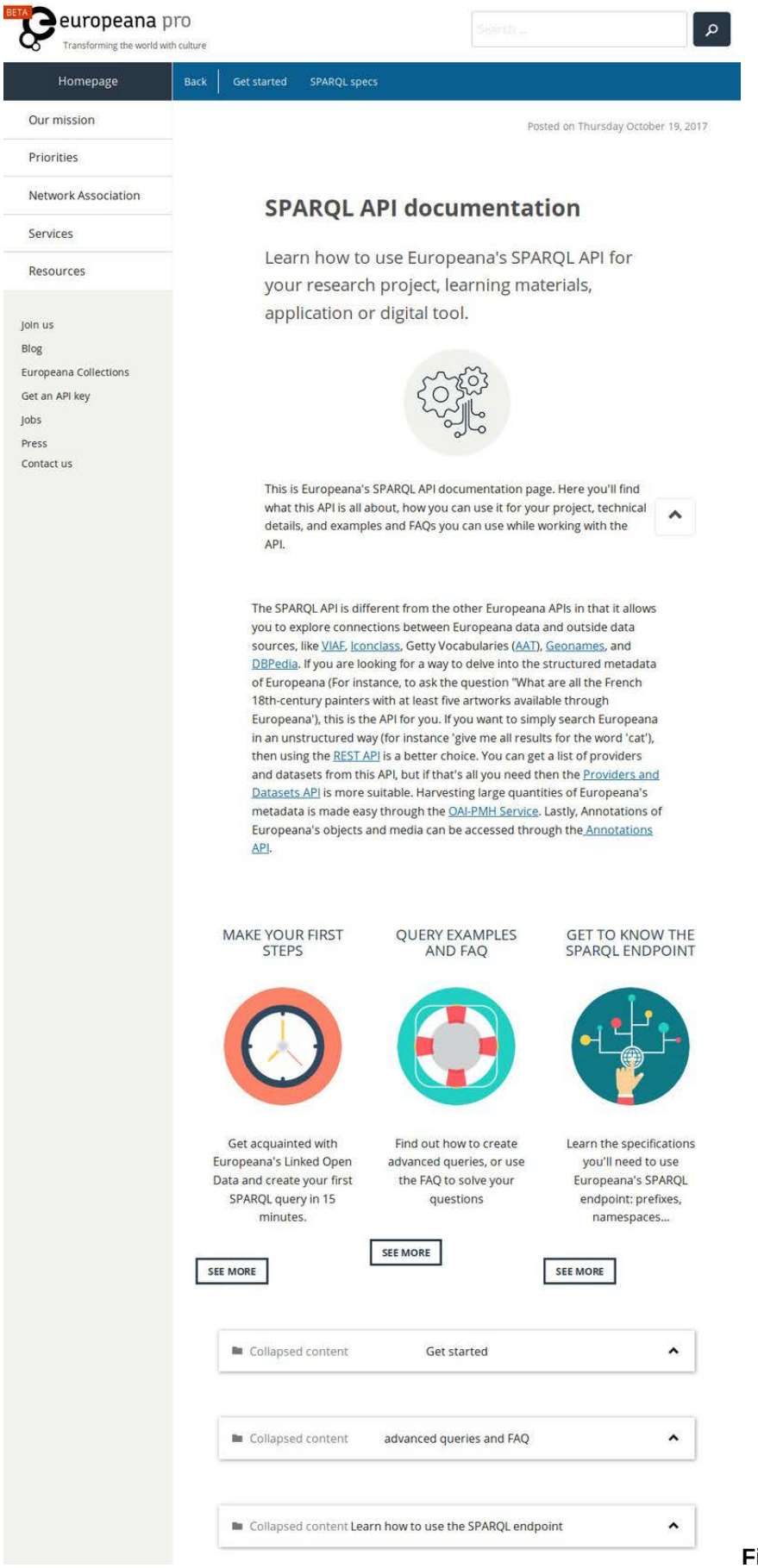

BETA
europeana pro
Transforming the world with culture
Search ...
Homepage
Back
Get started
SPARQL specs
Our mission
Priorities
Network Association
Services
Resources
Join us
Blog
Europeana Collections
Get an API key
Jobs
Press
Contact us
Posted on Thursday October 19, 2017
SPARQL API documentation
Learn how to use Europeana's SPARQL API for your research project, learning materials, application or digital tool.
This is Europeana's SPARQL API documentation page. Here you'll find what this API is all about, how you can use it for your project, technical details, and examples and FAQs you can use while working with the API.
The SPARQL API is different from the other Europeana APIs in that it allows you to explore connections between Europeana data and outside data sources, like VIAF, Iconclass, Getty Vocabularies (AAT), Geonames, and DBPedia. If you are looking for a way to delve into the structured metadata of Europeana (For instance, to ask the question "What are all the French 18th-century painters with at least five artworks available through Europeana'), this is the API for you. If you want to simply search Europeana in an unstructured way (for instance 'give me all results for the word 'cat'), then using the REST API is a better choice. You can get a list of providers and datasets from this API, but if that's all you need then the Providers and Datasets API is more suitable. Harvesting large quantities of Europeana's metadata is made easy through the OAI-PMH Service. Lastly, Annotations of Europeana's objects and media can be accessed through the Annotations API.
MAKE YOUR FIRST STEPS
QUERY EXAMPLES AND FAQ
GET TO KNOW THE SPARQL ENDPOINT
Get acquainted with Europeana's Linked Open Data and create your first SPARQL query in 15 minutes.
Find out how to create advanced queries, or use the FAQ to solve your questions
Learn the specifications you'll need to use Europeana's SPARQL endpoint: prefixes, namespaces...
SEE MORE
SEE MORE
SEE MORE
Collapsed content Get started
Collapsed content advanced queries and FAQ
Collapsed content Learn how to use the SPARQL endpoint

Fig 1.3.

The Europeana API documentation does say a lot about what is important to include for users with high technical expertise. Terms of use, deprecation information, a roadmap and changelog, information about backwards and forwards compatibility, datatypes for metadata fields, error response codes, faceting, pagination, query syntax, and libraries and plugins are important to include in your documentation. Europeana offers a variety of libraries and plugins. There is also a discussion forum for users of the API, and an email address for feedback and questions. There is no direct call to action to users to contact Europeana if they have created something new with the API though. Europeana's API architecture includes important metadata fields dealing with copyright restrictions. The 'reusability' parameter gives a clear indication of which objects can be reused, and under which circumstances. This is a crucial tool for creative users and researchers, who are often looking for objects that comply with specific reuse guidelines. Openness of digital cultural heritage is important, but often it is simply not possible to have all digital cultural heritage objects be at the same level of openness or reusability. In that case, clearly stating which objects are reusable to what degree allows users to find objects corresponding to the level of openness they're looking for.

To conclude, Europeana Pro's new API documentation attempts at incentivizing different groups of users to explore and create with Europeana's data using the Europeana APIs. The move to an integrated platform, pro.europeana.eu, is a great first step towards this goal. The actual documentation pages will need work to be able to convey the complexity of Europeana's API architecture and its data model in a simple way. There are already features in place to increase understandability, such as an API console, example calls, and libraries and plugins. Short-term opportunities for improving the documentation would be the creation of tutorials, ready-made code snippets, and FAQs.

Conclusion

This research paper aimed to capture the diversity of access to data in todays' diverse landscape of digital cultural heritage. CHI's are in different states of openness and accessibility when it comes to digital cultural heritage, and they regularly face decisions on how to provide access to the cultural heritage objects in their possession. Some release their data as downloadable datasets. Others act as a platform for reuse and development by providing access through an API. Even others close off their digital cultural heritage collection and monetise it, insofar as this is possible within their jurisdiction. I chose not to use the latter category as an example, since the purpose of this thesis is to provide a range of feasible options for sustainable access and sharing of cultural heritage. I do not believe that imposing copyright restrictions on digital cultural heritage and monetising the dissemination of those objects is a feasible option for sustainable access and sharing of cultural heritage.

Who are the users of CHI APIs?

One of the leitmotifs of the literature study and case studies of digital cultural heritage institutions is the broad spectrum of users a CHI's API markets itself to. This is often in contrast to other commercial APIs, that are specifically designed to appeal to developers with high technical expertise and knowledge. It is exactly the differing levels of technical expertise that makes creating and presenting CHI APIs more intricate than other APIs. Secondly, there is a will to motivate users to reuse and repurpose the content that CHIs provide. CHIs often go through a lot of effort to make their digital content available to users, and then expect that the quote 'If you build it, they will come' will ring true. There is a definite need to not only build it, but to also market it to users and guide them through a pleasant user experience. Having a complex API structure allows for much more possibilities offered to the user in their API calls, like fine granularity adjustments of queries. The more complex an API structure is, however, the harder it might be to explain it clearly to new users. Finding a balance between easy, intuitive querying and a concise explanation of the parameters versus having a complex and innovative API that uses interoperable metadata standards isn't easy. Techniques like the possibility of combining fields, and introducing users to the complexity of an API in stages, can help CHIs find that balance.

Next to having to cater to users with different levels of technical expertise, CHI APIs are used by users with varying motivations. Cultural Heritage is interesting for Research, for creative reuse, for application development… Offering different options of access to cultural heritage data allows CHIs to attract this broad range of users. Offering downloadable datasets, even if there is an API available, might benefit digital humanists who are looking for ready-made cleaned up datasets. Providing code snippets for popular applications of the API might benefit creative users and end-users.

Users of APIs are a complex group of people with differing levels of technical expertise and with varying motivations. Researchers, educators, developers, end-users, culture lovers, creative users, are all valid user groups to cater an API towards.

Why is thinking about how to provide access to digital cultural heritage important for CHIs?

CHIs make design choices in the development of their digital services. Throughout the literature study of this thesis, a few arguments were raised that influence these design choices. The Harvard Art Museums described general reservations quite some CHIs uttered about opening up digital collections to the public. One of the risks associated with openness is the loss of control over data. Only providing a portal into digital cultural heritage may, however, give a false sense of security to CHIs. Opening up digital collections and giving access in a controlled way allows for much more control of data than only providing a portal, which may be scraped and reused anyway.

Making cultural digital heritage accessible through an API allows a CHI to become more than just a service, or a portal to cultural heritage. It opens up the possibility of reuse, feedback from users, new and innovative applications that the CHI might have never thought of. Becoming a platform for digital cultural heritage is an attractive step for CHIs. On top of that, when publishing data that is Linked and Open, the opportunity arises of embedding a CHIs digital cultural heritage within the larger context of semantic cultural heritage data. Connecting to other CHIs and their data enriches everyone's objects, and fosters cross-institutional collaboration.

It's important for CHIs to gauge the impact of their services in some way, to see how they can improve their own work. APIs and their presentation and documentation pages offer the possibility of gaining more metrics to track the impact of a CHIs objects through use and reuse. People with no connection to the CHI at all that decide to build something creative, new, and inspiring on top of a CHIs platform is quite a rewarding return on investment for the institution. CHIs also have the power to foster innovation by opening up their data to new developments in research. Academics of Machine Learning are always on the lookout for usable data, and a CHI can be an ideal provider. Lastly, CHIs might invest in futureproofing their collections by publishing their content as Linked Open Data, connected to the semantic web.

Whether a CHI decides to make their data accessible or not, the ways in which they do so entails choices that impact the connection between users and institutions. It's important to make these design choices consciously, so they can reflect the goals and policies of a cultural heritage institution.

How can CHI's appeal to users? Suggestions for Cultural Heritage Institutions

Finally, analysing the several ways in which CHIs offer their data to users online, and describing some of them as use cases, has culminated in a list of suggestions for CHIs on how to present their open data, through an API or otherwise. These suggestions are linked to the kinds of users a CHI want to attract, since different users require different approaches. This is by no means an exhaustive list, but it might help CHIs put in perspective how qualitative their API documentation is, and what they can do to further improve the presentation of their data.

For users with low technical expertise:

- Provide step-by-step tutorials that don't require coding to introduce users to the architecture of an API
- Make sure parameters and field descriptions are accompanied by relevant examples
- Create an API console so users can experiment directly with their newfound knowledge of API usage.
- Organise training opportunities for in-house staff to let them become familiar with technical aspects, so they can help others when asked
- Make sure tutorials and examples are relevant to the users you want to attract, so they have a starting point to build on
- Provide ready-made code snippets for the most general use cases of your data, e.g. a simple embeddable search widget.
- Take the possibilities of your APIs with you to other public outreach projects, be if for education, research, family history enthusiasts, etc.

For users with high technical expertise

- Make detailed documentation pages searchable
- Provide info about error codes, HTML responses, and debugging info
- Answer advanced questions and issues in a technical FAQ
- Make it possible for users to help each other and discuss the use of the API in a public online discussion forum

For researchers:

- Foster federated access points so different APIs can be used together. Participate in publishing linked open data for the semantic web.
- Provide downloadable datasets for humanities computing, even if you also provide an API.
- Offer responses in more than one format, preferably both XML and JSON.
- Publish the internal statistical analysis you did on your data, to show preliminary results about the contents of the repository.
- Promote researchers' publications and projects when it uses and/or analyses your content

For creative users:

- Create funding opportunities for reuse of cultural heritage through your APIs
- Organise hackathons and other dissemination activities to encourage reuse
- Encourage user feedback, tracking impact of the API through what users have created on your platform
- Provide clear rights statements for every object you provide, so creative users know if and how they can reuse your data.

For developers:

- Document how your main portal and your other internal applications built on the API are created
- Consider making the development of internal applications and portals open-source, and provide links to the source code of your internally built services
- Consider offering the possibility of archiving, or even hosting, projects that have been built on your platform, to improve the sustainability and long-term preservation of external projects.
- Provide version history for developers tasked with maintaining projects dependent on the API.
- Provide Software Development Kits (SDKs) and code libraries, so developers can code with the API in the programming language they prefer.
- Clearly state Terms of Use for users that might consider marketing or monetising the projects they build on top of your platform.

Bibliography and Further Reading

Adeyinka, Tella, and Kwanya Tom. *Handbook of Research on Managing Intellectual Property in Digital Libraries*. IGI Global, 2017.

Buranasing, Watchira, Marut Buranarach, and Chai Wutiwiwatchai. “Open Data Integration for Lanna Cultural Heritage E-Museums,” 29–41. Advances in Intelligent Systems and Computing. Springer, Cham, 2015. https://doi.org/10.1007/978-3-319-70019-9_3.

Concordia, Cesare, Stefan Gradmann, and Sjoerd Siebinga. “Not Just Another Portal, Not Just Another Digital Library: A Portrait of Europeana as an Application Program Interface.” *IFLA Journal* 36, no. 1 (March 1, 2010): 61–69. https://doi.org/10.1177/0340035209360764.

Damova, Mariana, and Dana Dannells. “Reason-Able View of Linked Data for Cultural Heritage.” In *Third International Conference on Software, Services and Semantic Technologies S3T 2011*, 17–24. Advances in Intelligent and Soft Computing. Springer, Berlin, Heidelberg, 2011. https://doi.org/10.1007/978-3-642-23163-6_3.

Doerr, Martin, Crete FORTH-ICS, Gerald de Jong, and B. V. Delving. “Realizing Lessons of the Last 20 Years: A Manifesto for Data Provisioning & Aggregation Services for the Digital Humanities (a Position Paper).” *D-Lib Magazine* 20, no. 7/8 (2014).

Dole, Wanda V., Jitka M. Hurych, and Wallace C. Koehler. “Values for Librarians in the Information Age: An Expanded Examination.” *Library Management* 21, no. 6 (August 1, 2000): 285–97. https://doi.org/10.1108/01435120010327597.

Harvard. “Free for All | Index Magazine | Harvard Art Museums.” Accessed November 21, 2017. https://www.harvardartmuseums.org/article/free-for-all.

Hicks, Deborah. “Ethics in the Age of Technological Change and Its Impact on the Professional Identity of Librarians.” *Http://Services.Igi-Global.Com/Resolvedoi/Resolve.Aspx?Doi=10.4018/978-1-4666-8195-8.Ch071*, 2015, 1459–79. https://doi.org/10.4018/978-1-4666-8195-8.ch071.

Koehler, Wallace. *Ethics and Values in Librarianship: A History*. Rowman & Littlefield, 2015.

Ledesma, Andrea. “How We Learned to Stop Worrying and Love Open Data: A Case Study in the Harvard Art Museums’ API.” Https://medium.com/@andrea_ledesma. Andrea Ledesma on Medium, 2016. https://medium.com/@andrea_ledesma/how-we-learned-to-stop-worrying-and-love-open-data-a-case-study-in-the-harvard-art-museums-api-893c3f40ecb7.

Llamas, Jose, Pedro M. Lerones, Eduardo Zalama, and Jaime Gómez-García-Bermejo. "Applying Deep Learning Techniques to Cultural Heritage Images Within the INCEPTION Project," 25–32. Lecture Notes in Computer Science. Springer, Cham, 2016. https://doi.org/10.1007/978-3-319-48974-2_4.

Marden, Julia, Carolyn Li-Madeo, Noreen Whysel, and Jeffrey Edelstein. "Linked Open Data for Cultural Heritage: Evolution of an Information Technology," 107–112. ACM, 2013.

Markhoff, Béatrice, Thanh Binh Nguyen, and Cheikh Niang. "When It Comes to Querying Semantic Cultural Heritage Data," 384–94. Communications in Computer and Information Science. Springer, Cham, 2017. https://doi.org/10.1007/978-3-319-67162-8_38.

"Museums and the Machine-Processable Web / Cool Stuff Made with Cultural Heritage APIs." Accessed November 21, 2017. http://museum-api.pbworks.com/w/page/21933412/Cool%20stuff%20made%20with%20cultural%20heritage%20APIs.

Peter, Eklund, Wray Tim, and Ducrou Jon. "Linking Objects and Their Stories: An API For Exploring Cultural Heritage Using Formal Analysis." *JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE* 3, no. 3 (2011): 239–52. https://doi.org/10.4304/jetwi.3.3.239-252.

"Reprogramming The Museum | Museumsandtheweb.Com." Accessed December 20, 2017. https://www.museumsandtheweb.com/mw2011/papers/reprogramming_the_museum.

Sherratt, Tim. "Have Collection, Will Travel," January 4, 2014. http://discontents.com.au/have-collection-will-travel/.

Tim, Sherratt. "FROM PORTALS TO PLATFORMS: BUILDING NEW FRAMEWORKS FOR USER ENGAGEMENT." Hamilton, New Zealand, 2013. https://www.nla.gov.au/our-publications/staff-papers/from-portal-to-platform.

Waibel, Günter, Ralph LeVan, and Bruce Washburn. "Museum Data Exchange: Learning How to Share." *D-Lib Magazine* 16, no. 3/4 (2010). https://doi.org/10.1045/march2010-waibel.

"Why Build an API for a Museum Collection?" Accessed December 20, 2017. https://www.sfmoma.org/why-build-api-museum-collection/.

Yasser, A., Kathy Clawson, Chris Bowerman, and M. Lévêque. "Saving Cultural Heritage with Digital Make-Believe: Machine Learning and Digital Techniques to the Rescue." Sunderland University, St Peter's Campus, 2017. http://hci2017.bcs.org/.

Annex 1: DigitalNZ website screenshots

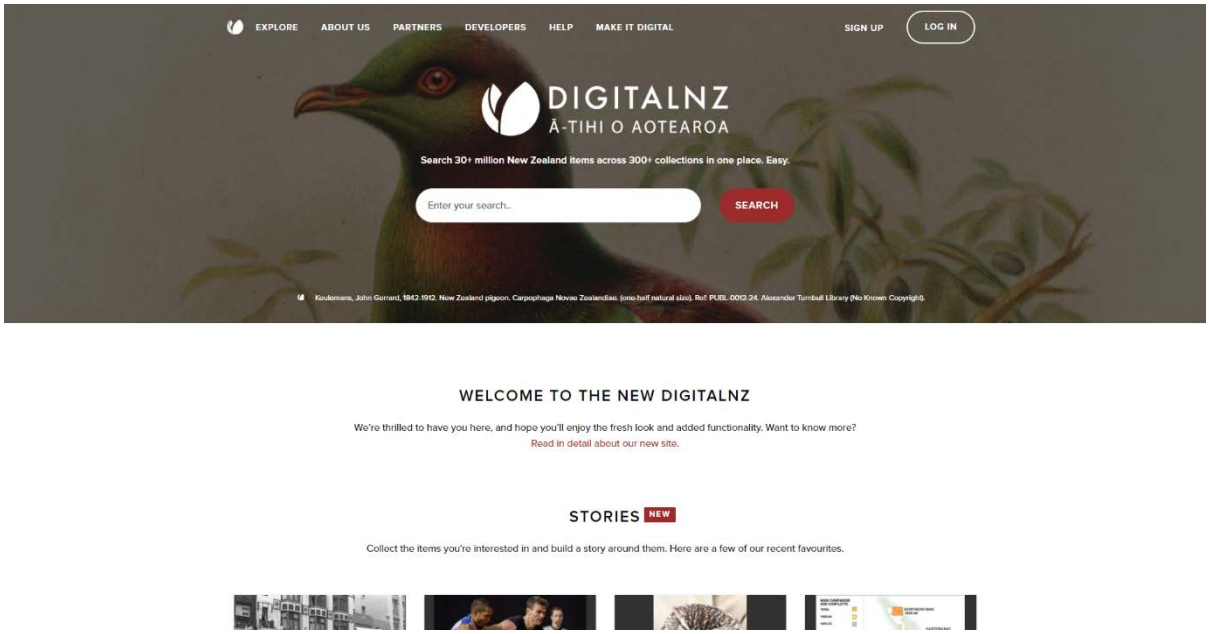


Fig 2.1. DigitalNZ homepage . https://digitalnz.com

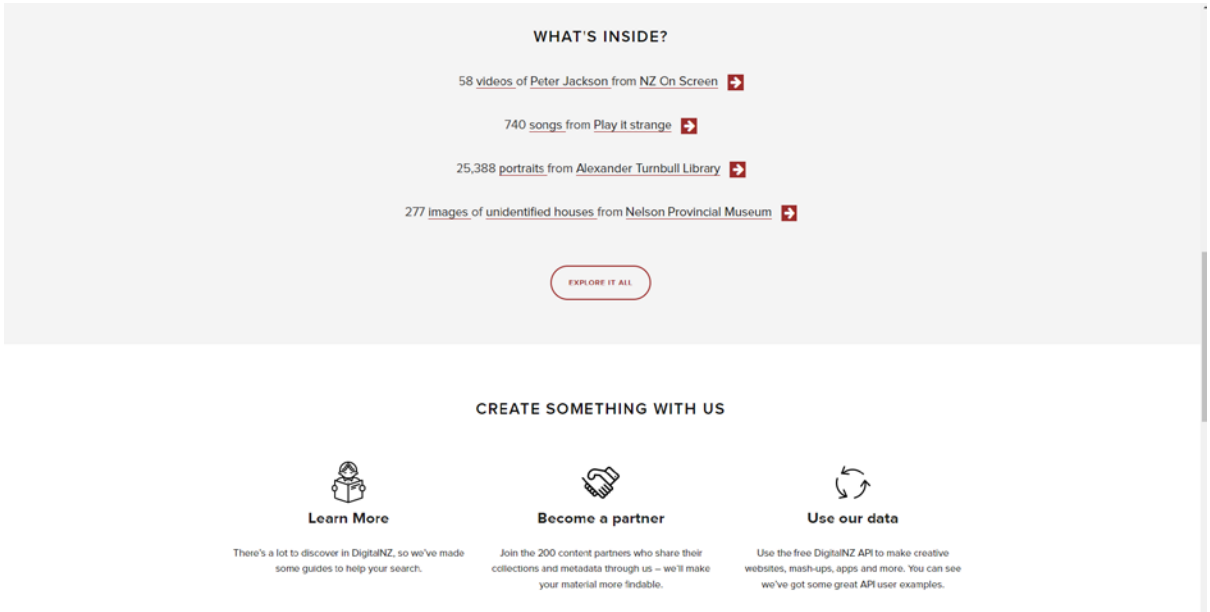


Fig 2.2. DigitalNZ homepage: under the fold

EXPLORE ABOUT US PARTNERS DEVELOPERS HELP MAKE IT DIGITAL SIGN UP LOG IN

GETTING STARTED
API EXAMPLES IN USE
API DOCS V3
CODE SAMPLES AND LIBRARIES
SUPPLEJACK

DEVELOPERS

USE THE DIGITALNZ API TO ACCESS FILES FROM ACROSS THE NZ CULTURAL, EDUCATION, AND GOVERNMENT SECTORS

The data available through DigitalNZ describes the digital items held by content partners across all walks of NZ life. You can use the API to search across the likes of titles, descriptions, dates, and creators of the material; as well as requesting thumbnails and the full metadata record for each item.

The aggregation service collects only the metadata that describes the digital content, but these API also return pointers to the item objects and thumbnails put online by our fantastic content partners. The Developer API Terms of Use explain how this all works. We hope you'll use this metadata to do good and wonderful things, create new experiences, and demonstrate the value of sharing this data. Email us at develop@digitalnz.org if we can help.

You can find examples of how others have used the API in our API examples showcase.

The current version of the DigitalNZ API is version 3.

- Sign-up and get your API key
- Getting started
- API documentation
- DigitalNZ Metadata Dictionary
- Code samples and libraries
- Developer API Terms of Use
- Ask a question on the Google group of DigitalNZ developers

Fig 2.3. DigitalNZ API landing page, https://digitalnz.org/developers/

GETTING STARTED
API EXAMPLES IN USE
API DOCS V3
CODE SAMPLES AND LIBRARIES
SUPPLEJACK

API EXAMPLES IN USE

DIGITALNZ API INTEGRATIONS SHOWCASE

The DigitalNZ API enables developers to find and query data from across the New Zealand cultural, education, and government sectors and create to new digital experiences. Here are some of the things people have made using our API. If you think we've missed your example of using the DigitalNZ API c ontact us and we may include it here.

National Library of New Zealand Website

The DigitalNZ API powers the search experience on the National Library website. It enables visitors to search across items in the various National Library of New Zealand catalogues as well as the Alexander Turnbull Library's unpublished collections.

N4L Pond

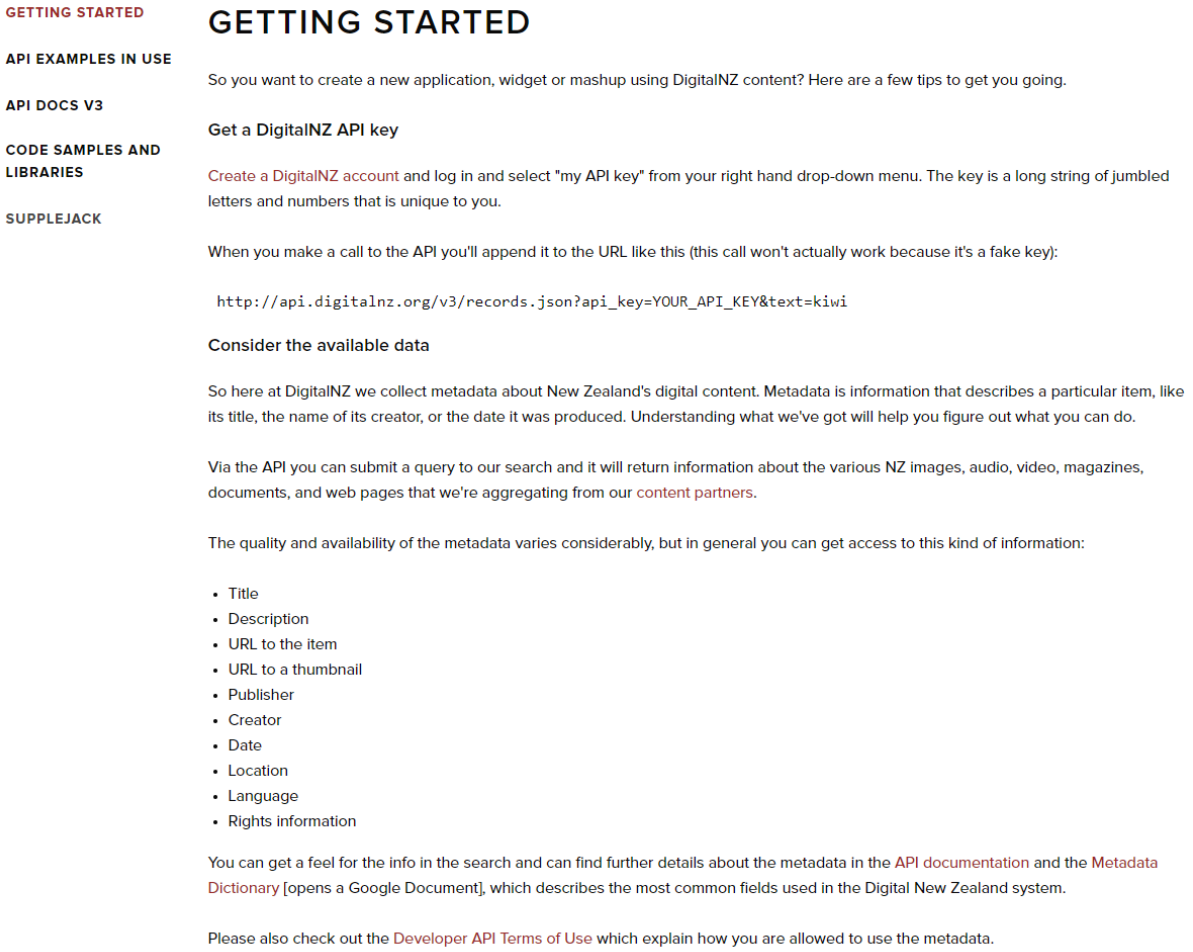
Pond, the Network for Learning's portal, is an online environment for education resource discovery and participation. Pond uses the DigitalNZ API to search across hundreds of institutions for relevant digital content.

Introducing Pond, the Network for Learning portal

CEISMIC

The Canterbury Earthquake Digital Archive (CEISMIC) is a comprehensive digital archive of video, audio, documents and images related to

Fig 2.4. DigitalNZ API examples page, https://digitalnz.org/developers/api-examples-in-use



GETTING STARTED

API EXAMPLES IN USE

API DOCS V3

CODE SAMPLES AND LIBRARIES

SUPPLEJACK

GETTING STARTED

So you want to create a new application, widget or mashup using DigitalNZ content? Here are a few tips to get you going.

Get a DigitalNZ API key

Create a DigitalNZ account and log in and select "my API key" from your right hand drop-down menu. The key is a long string of jumbled letters and numbers that is unique to you.

When you make a call to the API you'll append it to the URL like this (this call won't actually work because it's a fake key):

```
http://api.digitalnz.org/v3/records.json?api_key=YOUR_API_KEY&text=kiwi
```

Consider the available data

So here at DigitalNZ we collect metadata about New Zealand's digital content. Metadata is information that describes a particular item, like its title, the name of its creator, or the date it was produced. Understanding what we've got will help you figure out what you can do.

Via the API you can submit a query to our search and it will return information about the various NZ images, audio, video, magazines, documents, and web pages that we're aggregating from our content partners.

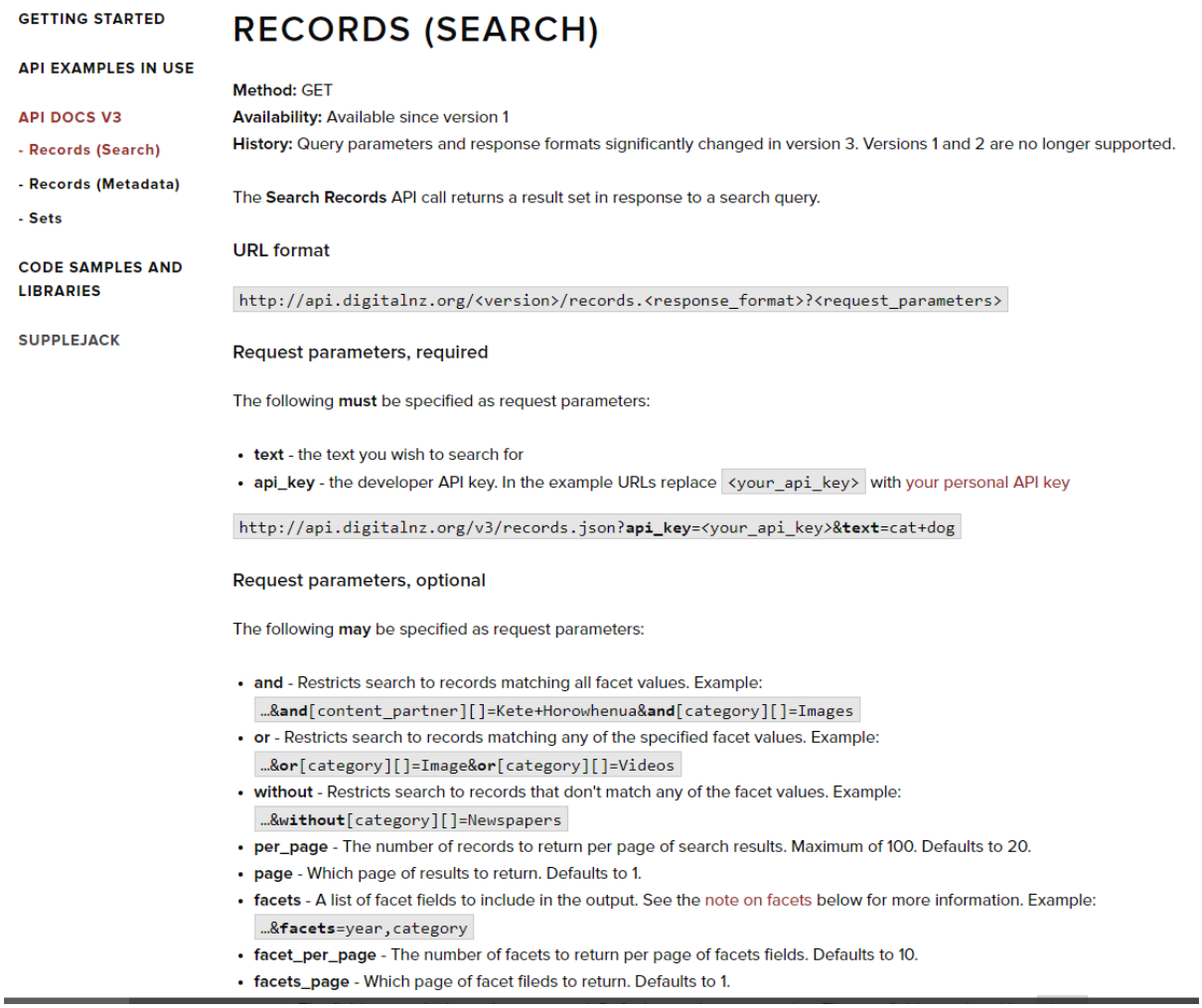
The quality and availability of the metadata varies considerably, but in general you can get access to this kind of information:

- Title
- Description
- URL to the item
- URL to a thumbnail
- Publisher
- Creator
- Date
- Location
- Language
- Rights information

You can get a feel for the info in the search and can find further details about the metadata in the API documentation and the Metadata Dictionary [opens a Google Document], which describes the most common fields used in the Digital New Zealand system.

Please also check out the Developer API Terms of Use which explain how you are allowed to use the metadata.

Fig 2.5. DigitalNZ API Getting Started page, https://digitalnz.org/developers/getting-started



GETTING STARTED

API EXAMPLES IN USE

API DOCS V3

- Records (Search)
- Records (Metadata)
- Sets

CODE SAMPLES AND LIBRARIES

SUPPLEJACK

RECORDS (SEARCH)

Method: GET
Availability: Available since version 1
History: Query parameters and response formats significantly changed in version 3. Versions 1 and 2 are no longer supported.

The **Search Records** API call returns a result set in response to a search query.

URL format

```
http://api.digitalnz.org/<version>/records.<response_format>?<request_parameters>
```

Request parameters, required

The following **must** be specified as request parameters:

- **text** - the text you wish to search for
- **api_key** - the developer API key. In the example URLs replace `<your_api_key>` with your personal API key

```
http://api.digitalnz.org/v3/records.json?api_key=<your_api_key>&text=cat+dog
```

Request parameters, optional

The following **may** be specified as request parameters:

- **and** - Restricts search to records matching all facet values. Example:
  `…&and[content_partner][]=Kete+Horowhenua&and[category][]=Images`
- **or** - Restricts search to records matching any of the specified facet values. Example:
  `…&or[category][]=Image&or[category][]=Videos`
- **without** - Restricts search to records that don't match any of the facet values. Example:
  `…&without[category][]=Newspapers`
- **per_page** - The number of records to return per page of search results. Maximum of 100. Defaults to 20.
- **page** - Which page of results to return. Defaults to 1.
- **facets** - A list of facet fields to include in the output. See the note on facets below for more information. Example:
  `…&facets=year,category`
- **facet_per_page** - The number of facets to return per page of facets fields. Defaults to 10.
- **facets_page** - Which page of facet fileds to return. Defaults to 1.

Fig 2.6. digitalNZ Search API documentation page, https://digitalnz.org/developers/api-docs-v3/search-records-api-v3

Annex 2 : Europeana Pro screenshots

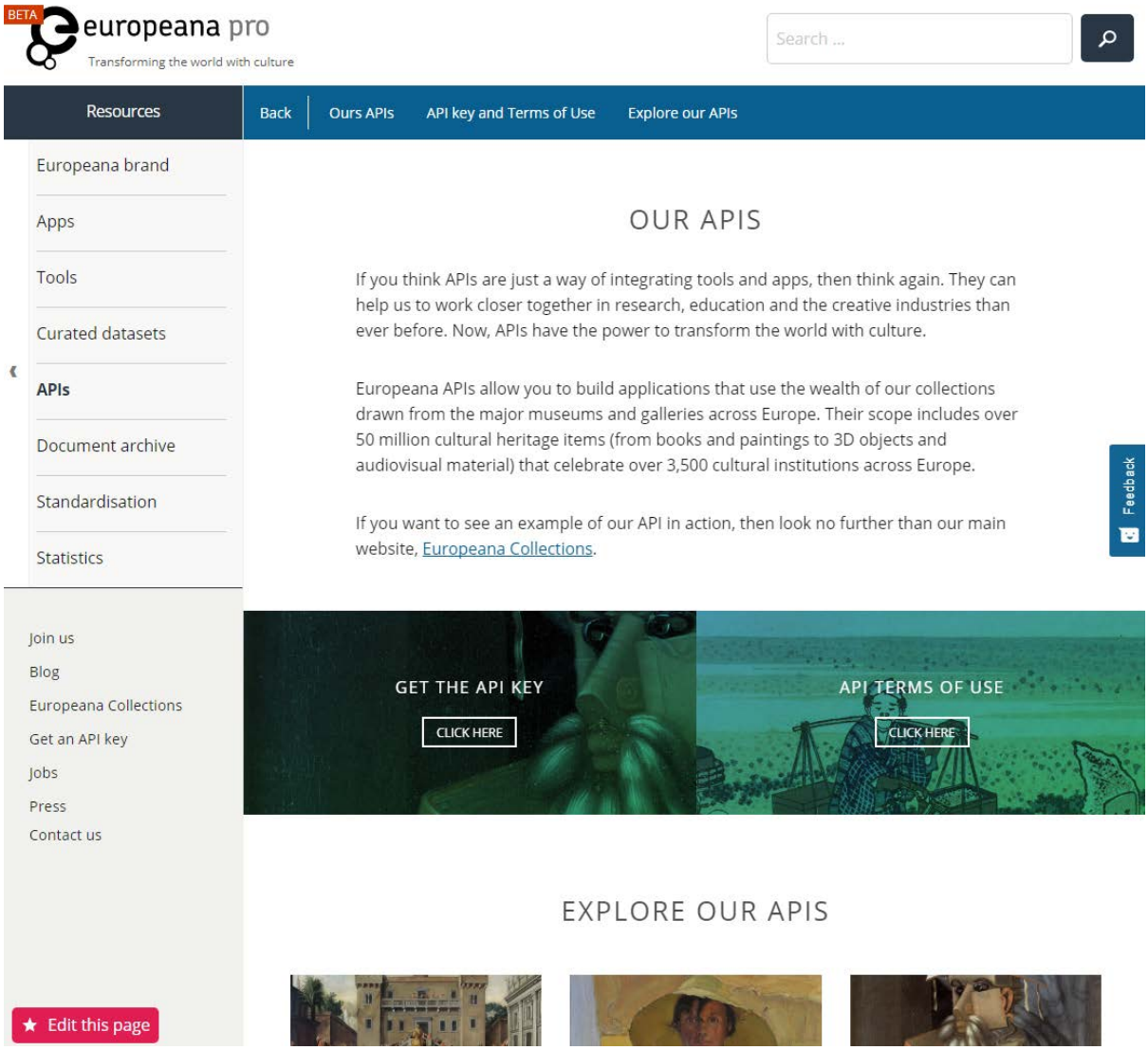


Fig 3.1. The Europeana APIs landing page

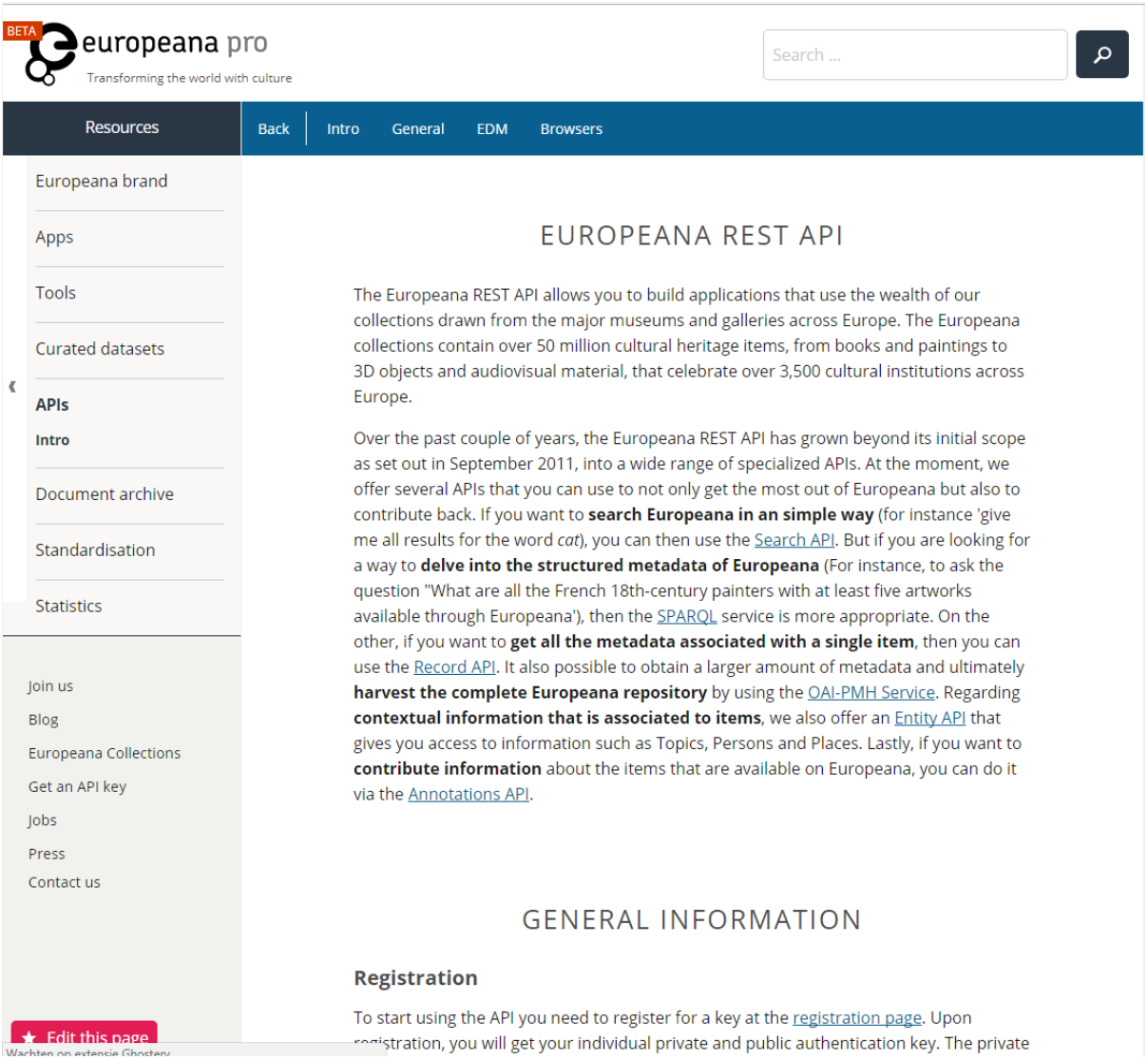


Fig 3.2. The Introduction to Europeana APIs

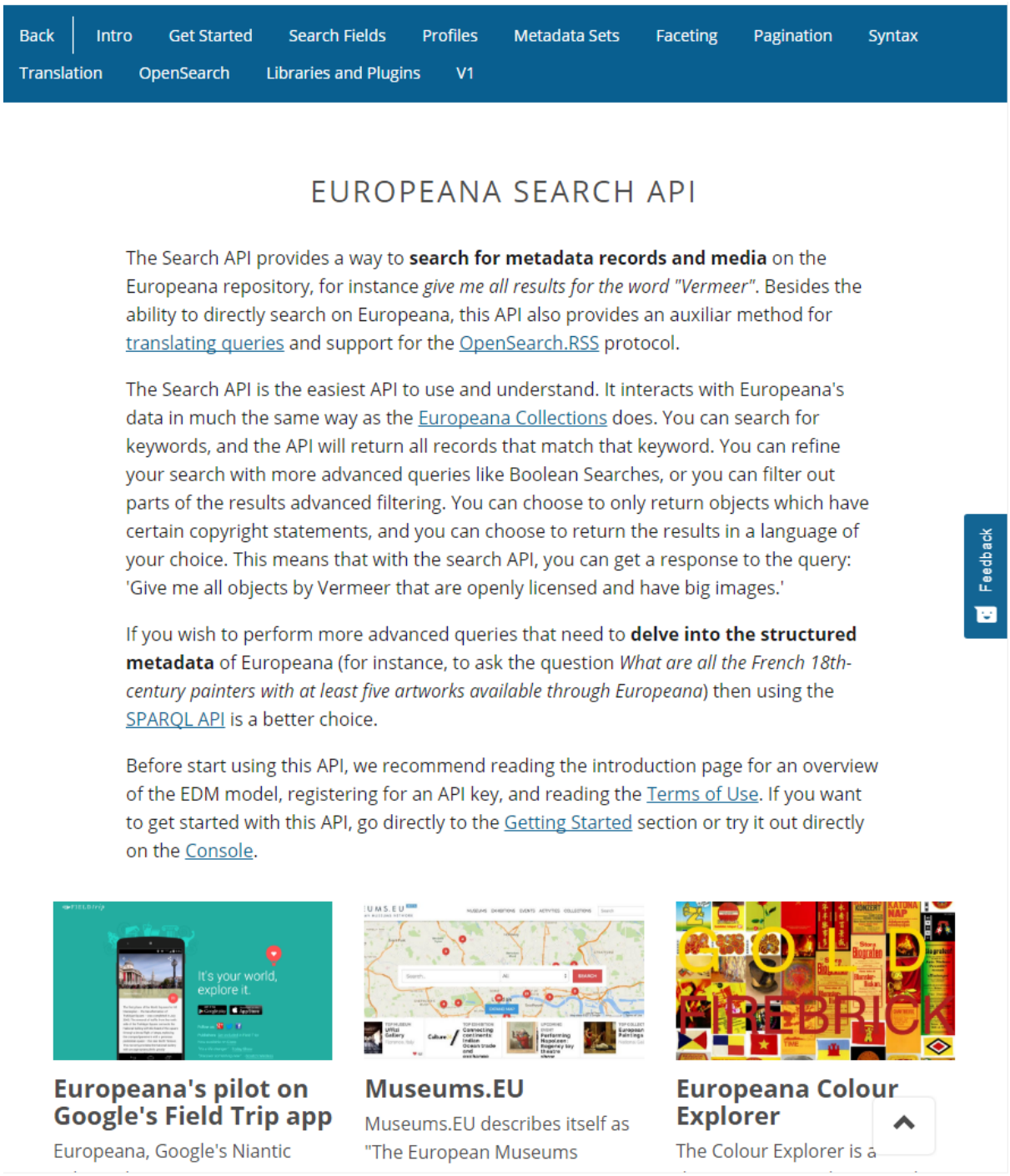


Fig 3.3. Example of a Europeana API documentation page: the Europeana Search API documentation