

CERN

SUMMER STUDENT REPORT

Working for Digital Library Services

Author:
Ivi DIMOPOULOU

Supervisor:
Charalampos
TZOVANAKIS

August 21, 2015



Abstract

The present report intends to briefly outline the work that has been conducted by me, as part of the Summer Student Program (Member States), which I attended during the summer of 2015.

Contents

1	Definitions	3
1.1	CERN Document Server (CDS)	3
1.2	Invenio	3
2	Introduction	3
3	Main task	3
3.1	What I used for the task	4
3.2	The stages of the development process	4
3.2.1	The learning stage	4
3.2.2	Defining what needs to be done	5
3.2.3	Mockups	5
3.2.4	The functionality of the brief views page	5
3.2.5	The layout of the page	6
3.2.6	The dynamic components of the page	6
3.2.7	Examples	6
3.2.8	Testing	7
3.2.9	Review and alterations	8
3.2.10	Pull request	8
4	Supplementary task	9
5	Conclusions	9

1 Definitions

1.1 CERN Document Server (CDS)

The CERN Document Server or CDS is a server and also an electronic library of the *Conseil Européen pour la Recherche Nucléaire*(CERN). It has been managing over 1,000,000 bibliographic records in high-energy physics since 2002, covering articles, books, journals, photos, videos, and more.

1.2 Invenio

Invenio is a free software suite enabling the operation of a digital library or document repository on the web. The technology, offered by the software, covers all aspects of digital library management, from document ingestion through classification, indexing, and curation to dissemination. Invenio has been originally developed at CERN to run the CERN Document Server and is currently being co-developed by an international collaboration, comprising of institutes such as CERN, DESY, EPFL, FNAL and SLAC. It is being used by about thirty scientific institutions worldwide.

2 Introduction

During my participation in the Summer Student Program of CERN, for the year 2015, I was a member of the Digital Library Services (DLS) section of the IT Department. The DLS section manages the CERN Document Server, using technologies based on Invenio. The section is currently in the process of making a new, more functional, convenient and elegant version of the CERN Document Server.

3 Main task

The task which was assigned to me was to manufacture the brief view of each record in the CERN Document Server. To be more specific, a brief view is a first glance at a record. Therefore, it should cover small space and at the same time it should be as informative as the aforementioned space allows.

As any other web development project, the design and implementation of

the brief views needed the use of a number of different programming languages and tools. Furthermore, the project was divided into different stages which led to the production of the final result. The tools that were used and the stages of the process are briefly analyzed bellow.

3.1 What I used for the task

1. Programming languages: Python, HTML
2. Web development framework: Flask
3. Template engine: Jinja2
4. Styling tools: CSS, Bootstrap
5. Tools for the dynamic components of the page: Javascript, JQuery
6. Testing tool: Jasmine
7. Other pieces of software: Git

3.2 The stages of the development process

3.2.1 The learning stage

Before starting to work on a software development project, one should have the necessary knowledge of the programming languages, paradigms and pieces of software which are required for the implementation of the project. Especially in the case of a group project, which was my case, it is necessary for every new member of the group, to spend time in understanding and learning all the tools that are used by the group within the project.

In this context, the first weeks of my studentship here at CERN played the role of the learning stage for me. Searching the web for useful information and following tutorials, were both a big part of these weeks, due to the fact that I was bound to use technologies which I had never used before.

3.2.2 Defining what needs to be done

In every group project, this is the part where the team defines the work that has to be done, divides it into smaller tasks and thereafter assigns each task to the appropriate person. During this stage, I was assigned my task by my supervisor, I discussed with him thoroughly every aspect of it, and I started having a clearer picture of what my contribution to the project was going to be.

The task was split into two parts. The first one was the functional part, which contained the production of the code that would determine the actions which had to be executed during the loading of the brief views, in order for the page to be able to present the necessary information to the users. The second part was the part of the design, which aimed at making the page user-friendly, informative and elegant.

3.2.3 Mockups

In manufacturing and design, a mockup is a scaled or full-size model of a design or device, used for teaching, demonstration, design evaluation, promotion, and other purposes. It is often used in web development, in order to determine, with the help of an image rather than words, the shape and figure of web pages.

For the brief views, the DLS team decided to use the following design, as it is imprinted in the mockup bellow:

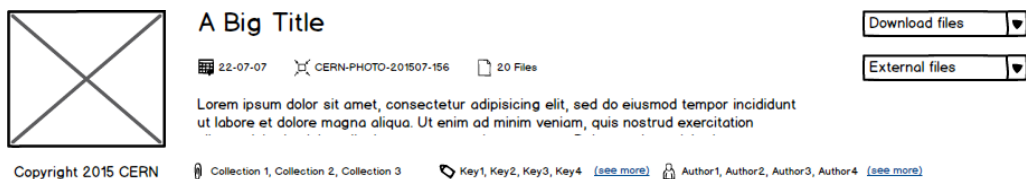


Figure 1: A schematic illustration of the brief view of a record in the CERN Document Server

3.2.4 The functionality of the brief views page

In order to ensure that the brief views page of the CERN Document Server would be able to provide the users with the necessary information, the pro-

duction of certain code segments was needed. The main tool that was used for this stage of the task, was the template engine Jinja2.

In particular, my main duty was to make a number of Jinja macros, that would play a significant role in the templating of the page.

3.2.5 The layout of the page

The basic tool that was used for the design of the page was the Hyper Text Markup Language (HTML). The static components of the page were easily constructed with the help of HTML, while CSS and, the front-end framework, Bootstrap, were involved wherever there was the need for decoration or special features.

3.2.6 The dynamic components of the page

For the dynamic content of the page, a programming language called Javascript was used. In particular, with the help of the JQuery library, I constructed a Javascript function which would allow the user to swap between a short and an expanded version of the abstract of a record. The aforementioned function is called "expandContent".

3.2.7 Examples

Below, an example of the brief view of a test record is displayed:



Figure 2: Brief view of a record in CDS

The functionality of the "expandContent" Javascript function is shown below:

3

High-order Large Eddy Simulations of Confined Rotor-Stator Flows

arXiv:1305.2885

In many engineering and industrial applications, the investigation of rotating turbulent flow is of great interest. In rotor-stator cavities, the centrifugal and Coriolis forces have a strong influence on the turbulence by producing a secondary flow in the meridian plane composed of two thin boundary layers along the disks ...[\(read more\)](#)

Article by [Viazzo Stéphane \(M2P2\)](#), [Poncet Sébastien \(M2P2\)](#) (see all) External Links

Figure 3: Short version of the abstract

3

High-order Large Eddy Simulations of Confined Rotor-Stator Flows

arXiv:1305.2885

In many engineering and industrial applications, the investigation of rotating turbulent flow is of great interest. In rotor-stator cavities, the centrifugal and Coriolis forces have a strong influence on the turbulence by producing a secondary flow in the meridian plane composed of two thin boundary layers along the disks separated by a non-viscous geostrophic core. Most numerical simulations have been performed using RANS and URANS modelling, and very few investigations have been performed using LES. This paper reports on quantitative comparisons of two high-order LES methods to predict a turbulent rotor-stator flow at the rotational Reynolds number $Re=400000$. The classical dynamic Smagorinsky model for the subgrid-scale stress (Germano et al., Phys Fluids A 3(7):1760-1765, 1991) is compared to a spectral vanishing viscosity technique (S'everac & Serre, J Comp Phys 226(2):1234-1255, 2007). Numerical results include both instantaneous data and postprocessed statistics. The results show that both LES methods are able to accurately describe the unsteady flow structures and to satisfactorily predict mean velocities as well as Reynolds stress tensor components. A slight advantage is given to the spectral SVV approach in terms of accuracy and CPU cost. The strong improvements obtained in the present results with respect to RANS results confirm that LES is the appropriate level of modelling for flows in which fully turbulent and transition regimes are involved. [\(read less\)](#)

Article by [Viazzo Stéphane \(M2P2\)](#), [Poncet Sébastien \(M2P2\)](#) (see all) External Links

Figure 4: Full version of the abstract

3.2.8 Testing

In a decent project, every new piece of code that has been produced has to be tested in terms of functionality and correctness, before its final release. In the case of my code, the Jasmine testing tool had to be used in order for the "expandContent" function to be thoroughly checked.

To be more specific, I designed a few tests (called specs in the terminology of Jasmine) which ensured that the function would work properly in every possible situation, without introducing an overhead to the overall time of execution of the code.

An example of the results of the tests for the "expandContent" function, is the following:

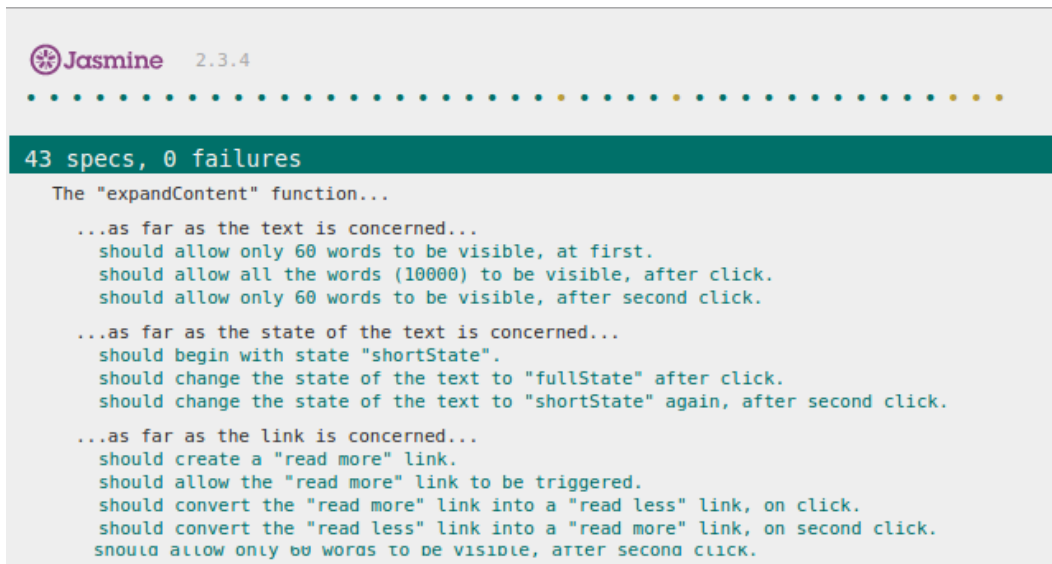


Figure 5: Eleven successful specs for the "expandContent" Javascript function

3.2.9 Review and alterations

A piece of code can be considered as "ready for release", only after it has been reviewed and approved by the appropriate members of the team, in order to ascertain that it follows the standards of the project, in terms of quality, layout and other criteria.

In the case of the DLS section, the review of any new code segment is performed through Git and Github. Therefore, after "pushing" my code on Github, I received a number of comments from other members of the section and I had to make a few alterations on the initial code.

3.2.10 Pull request

A "pull request" on Github is a request for the merge of someone's code segment to the group project. This was the final step of the process and it concluded my main task at CERN. After my pull request on Github, all that remains is to see if any part of it, will eventually be merged to the actual project.

4 Supplementary task

After finishing the biggest part of the work concerning my main task, I was assigned another, significantly smaller, task. In this task, the main goal was to avoid private information of restricted records in CDS to be obtained by external factors, such as web page translation services.

For the above reason, I manufactured another series of Jinja macros and Javascript functions, to ensure that an appropriate alert would appear at the top of a CDS page in an occasion such as the above.

The aforementioned behaviour is depicted in the following image:

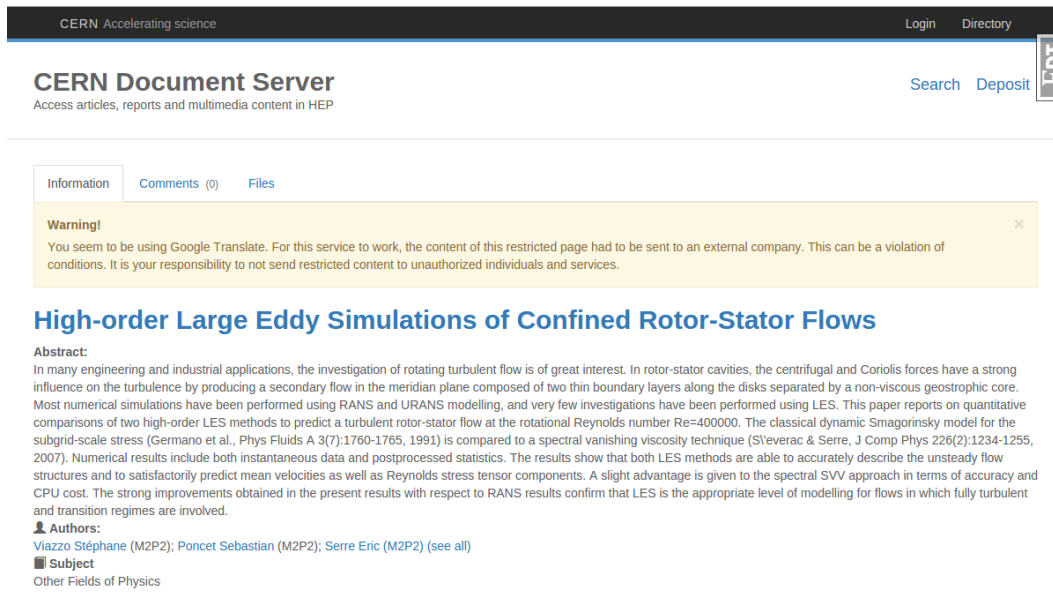


Figure 6: Alert in case of translation detection

5 Conclusions

All in all, this summer studentship at CERN has offered me a unique working experience. It gave me the opportunity to learn how to use a series of different tools, something that will definitely prove to be extremely useful in the future. Furthermore, by working as a member of such a big team and on a project

of proportional magnitude, I came to a number of conclusions, concerning software development. First of all, I re-discovered the value of cooperation and coordination between colleagues and team members. I also found that in order for someone to become a successful programmer, learning is a never-ending process. Finally, the most important thing that I have learned here is also a famous quote: "You never finish a project, you just stop working on it".

References

- [1] Invenio, <http://invenio-software.org>
- [2] Wikipedia - The free encyclopedia, <https://en.wikipedia.org>