Preliminary Project Plan for "Energy portal site 2.0 - API"

– version 2020-03-19a

Please note! The original ***project description*** is available at:
http://user.it.uu.se/~hesc0353/exjobb-dvk/EnergyPortal-2.0-API.pdf.
The original project *description* is ***not*** supposed to change.
This project plan is supposed to be *updated on a weekly basis or at least every second week*.

Title (preliminary)

Managing web API data sources

Background

The Uppsala Region is responsible, among other things, for the health care and public transport in Uppsala County. This includes keeping up the Academic Hospital, a socially important function with a life-sustaining purpose that must never stand still. The hospital is three blocks large, uses 100 GWh of energy annually (corresponding to about 7,000 villas). In order to work with the region's social impact, large energy-saving projects have been implemented in recent years and own electricity production has been introduced via solar cell installations comprising more than 1500 kW.

The Uppsala Region intends, through these installations, to also strengthen the innovation power in the Swedish solar industry. This is done by opening up data and disseminating the experience from the expansion of the solar plants, which are of many different technologies. The goal is to be able to compare them. Part of this work is the launch and further development of the site https://energiportalregionuppsala.se.

More data sources are needed for the portal and this is where this project comes in. Public organizations also have a legal requirement to disclose socially important data - the Public Sector Information Act. Very few have done this and here, Region Uppsala together with us at STUNS wants to make a difference.

The main goal of the project from the client's point of view is to develop solutions in an existing Azure environment to receive, store and manage data to and from primarily **two** web API data sources, known as the "FerroAmp" system and the "Save-by-solars" solar energy collectors.

Research objective (preliminary)

This study's objective has not been decided yet. In fact, the most important task for the **subject reviewer** to begin with, is to help me – Henrik Schulze – to formulate ***one*** research objective (at least ***one*** research question) that is suitable for a 15 credit bachelor thesis. The research question should be answered in the context of the technical environment at Region Uppsala – an Azure environment aiming at retrieving solar energy data from primarily two API data sources.
Please note! This work should be completed in just **half a semester** – ***no more than ten weeks***!

Research discipline and methodology (*Definitive?* – I hope so.)

Given the contextual environment at the company Region Uppsala – and the research interest from my side – I believe a suitable research discipline / methodology is ***design science***.
One classical reference for this methodology is [Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004).

Design science in information systems research. MIS Quarterly, 28 1), 75–105]. This article is available at https://wise.vub.ac.be/sites/default/files/thesis_info/design_science.pdf.

A more recent reference is https://wwwhome.ewi.utwente.nl/~roelw/DSM180minutes.pdf of 2016, and even more recently https://wwwhome.ewi.utwente.nl/~roelw/microtutorial.pdf of 2019.

Preliminary Research Question

The purpose of the study is to investigate how application program interfaces (APIs) should be managed to be as useful as possible, as well as how software can support this goal. Although the work on my thesis will depend heavily on input from the subject reviewer, here is one tentative research question:

What guidelines are essential for an efficient web API management?

More on the Subject Background (*very* preliminary)

- Six guiding constraints define a RESTful system.
  https://en.wikipedia.org/wiki/Representational_state_transfer#Architectural_constraints
- Cloud computing is a kind of Internet-based computing that offers pooled computer processing resources and data to computers and other devices on demand. It is often referred to as "the cloud" delivery of on-demand computer resources, everything from applications to data centers, through the Internet, normally, on a pay-for-use basis.
  https://apilama.com/2016/08/29/data-as-a-service-rest-apis-transforming-the-cloud-era/
- Cloud computing is the on-demand availability of computer system resources, especially data storage and [...] https://en.wikipedia.org/wiki/Cloud_computing
- Introduction to Cloud Computing by Jim Dowling
  https://www.kth.se/social/files/554fa451f276544829be2e5e/9-cloud-computing.pdf
- Introduction to CLOUD COMPUTING and OPENSTACK by Luis Tomás
  http://www8.cs.umu.se/kurser/5DV153/HT15/lectures/l10-handouts.pdf

Task Description (*very* preliminary)

It is important to stress that this study should be completed **within one single quarter of a year**. This is ***very*** different from spending four years on a PhD thesis.(!)

It is very hard – in fact, close to impossible – to write a task description before the research objective has been decided. Please see the *Research question / objective* section above.

From a *technical* (not necessarily academic) point of view, the following *type* of questions need to be answered satisfactorily:

- How do I get data from a REST API?
  - How to Parse JSON Data From a REST API Using a Simple JSON Library?
  https://dzone.com/articles/how-to-parse-json-data-from-a-rest-api-using-simpl

- How can data from different web API sources be combined into one single query?
- What is a JSON body?
  https://relishapp.com/apiary/gavel/docs/expectations/body-json-example
- How to create a unified REST API from a number of underlying APIs?
  https://docs.microsoft.com/en-us/azure/architecture/best-practices/api-design
- How to combine multiple REST API requests into a single request?
  https://stackoverflow.com/questions/26383797
- REST API Design: Multiple calls vs. single call to the API?
  https://softwareengineering.stackexchange.com/questions/341872

Relevant Literature and Reading List (likely to be extended)

- Roy Fielding's REST dissertation
  https://oleb.net/2018/rest/
- A view of cloud computing
  https://dl.acm.org/doi/fullHtml/10.1145/1721654.1721672
- Foster and Gannon. *Cloud Computing for Science and Engineering*. MIT Press. 2017.
- Design and development of a REST-based Web service platform for applications integration by LUIS OLIVA FELIPE
  https://upcommons.upc.edu/bitstream/handle/2099.1/8553/Master%20thesis%20-%20Luis%20Oliva.pdf
- Introducing collaborative Service Mashup design by Martin Vasko and Schahram Dustdar
  https://pdfs.semanticscholar.org/a92c/378b13ffcecc4cbc7b86edbada2be5e4acfc.pdf
- Six guiding constraints define a RESTful system.
  https://en.wikipedia.org/wiki/Representational_state_transfer#Architectural_constraints
- Cloud computing is a kind of Internet-based computing that offers pooled computer processing resources and data to computers and other devices on demand. It is often referred to as "the cloud" delivery of on-demand computer resources, everything from applications to data centers, through the Internet, normally, on a pay-for-use basis.
  https://apilama.com/2016/08/29/data-as-a-service-rest-apis-transforming-the-cloud-era/
- Cloud computing is the on-demand availability of computer system resources, especially data storage and [...] https://en.wikipedia.org/wiki/Cloud_computing
- Introduction to Cloud Computing by Jim Dowling
  https://www.kth.se/social/files/554fa451f276544829be2e5e/9-cloud-computing.pdf
- Introduction to CLOUD COMPUTING and OPENSTACK by Luis Tomás
  http://www8.cs.umu.se/kurser/5DV153/HT15/lectures/l10-handouts.pdf
- Cloud computing: Knowledge, expectation and reality by Andrea Rylander
  http://www8.cs.umu.se/kurser/5DV153/HT15/lectures/l10-handouts.pdf

- The Impact of Cloud Computing on IT Service Providers' Business Models by Trevor Clohessy https://aran.library.nuigalway.ie/bitstream/handle/10379/5620/2016clohessyphd.pdf
- RESTFUL SERVICES – APPLYING THE REST ARCHITECTURAL STYLE by Brian C. Henry http://www.aspensoft.com/resources/BrianHenryThesis-FinalDraft10-1-2011.pdf
- RESTful Web Services Development with a Model-Driven Engineering Approach by Rafael Gonçalves https://recipp.ipp.pt/bitstream/10400.22/11932/1/DM_RafaelGoncalves_2018_MEI.pdf
- API Design in Distributed Systems: A Comparison between GraphQL and REST by Thomas Eizinger, https://eizinger.io/assets/Master-Thesis.pdf

Preliminary systems, tools and methods are Windows OS, Visual Studio + Azure, RESTful APIs.

Relevant courses

The *most* relevant course is probably **Applied Cloud Computing**, https://studentportalen.uu.se/portal/portal/uusp/student/student-course?entityId=186251

Other relevant courses are those that relate to web technologies:

- **Website Construction** https://studentportalen.uu.se/portal/portal/uusp/student/student-course?entityId=102102
- **Web Design with HTML and CSS** https://studentportalen.uu.se/portal/portal/uusp/student/student-course?entityId=102467
- **System Design with a User Perspective** https://studentportalen.uu.se/portal/portal/uusp/student/student-course?entityId=162938

Additional relevant courses are those that contain object-oriented programming in Java and/or C#:

- **C# and Design of User Interfaces** https://studentportalen.uu.se/portal/portal/uusp/student/student-course?entityId=118264
- **Computer Networking for Programmers (Java)** https://studentportalen.uu.se/portal/portal/uusp/student/student-course?entityId=107132
- **Computer Programming I (Java)** https://studentportalen.uu.se/portal/portal/uusp/student/student-course?entityId=159840
- **Computer Programming II (Java)** https://studentportalen.uu.se/portal/portal/uusp/student/student-course?entityId=139106
- **Design Patterns (C#)** https://studentportalen.uu.se/portal/portal/uusp/student/student-course?entityId=118270
- **Information Systems C: Bachelor's Degree Project** https://studentportalen.uu.se/portal/portal/uusp/student/student-course?entityId=128779
- **Information Systems C: Object-Oriented Programming II (C#)** https://studentportalen.uu.se/portal/portal/uusp/student/student-course?entityId=131441
- **Introduction to Programming with Java** https://studentportalen.uu.se/portal/portal/uusp/student/student-course?entityId=102463
- **Specialised Programming with Java** https://studentportalen.uu.se/portal/portal/uusp/student/student-course?entityId=102465

***Possibly* relevant courses:**

- **Programming of Parallel Computers**
  https://studentportalen.uu.se/portal/portal/uusp/student/student-course?entityId=129250
- **Scientific Computing I**
  https://studentportalen.uu.se/portal/portal/uusp/student/student-course?entityId=121110
- **Scientific Computing II**
  https://studentportalen.uu.se/portal/portal/uusp/student/student-course?entityId=121112

^^ * Please note * ^^

Except for the course on Applied Cloud Computing, I have taken ***all*** of the above courses.

Limitations

This project is entirely focused on how to receive, store and manage data from and to web APIs. Apart from the research question (which has not been decided yet), nothing else will be in focus. This project concerns only the backend - no frontend will be involved. For example, there will ***not*** be any presentation of data on the web, based on some user's choice on the energy portal site. A completely different project already exists for that purpose.

Timetable (*very* preliminary)

What follows is an approximate division of time to be spent on various activities:

- Study period, APIs, Azure environment. Understanding the two data sources – "FerroAmp" and "Save-by-solars". ~2 weeks.
- Then work on an approximate solution / start coding. Connect to APIs, JSON data. ~1 week.
- Implementing how to read data from the API. How should the data be retrieved into the portal? ~3 weeks.
- Is there a more general approach that allows for retrieving data from *all* types of API sources – also those that have other formats? ~2 weeks.
- Completing the thesis report. ~2 weeks.

Summing up all of the above, this is a plan for ten weeks.

I have no idea when the project can start, since that is entirely dependent on how long it takes to find a subject reviewer. Consequently, neither can I set a date for when the thesis should be expected to have been completed.

How to read this project plan

Intended audiences for the different parts of this project plan are as follows.

Text written in **black** is intended for *anyone* who takes an interest in reading this plan.

Text written in **red** is primarily intended for *the supervisor*.

Text written in **green** is primarily intended for *the subject reviewer*.

The thesis coordinator and the teacher(s?) on the course are expected to read **everything** in this project plan.

Once I have been registered on the course, the intended audience of the **updated versions** of this plan will be **the subject reviewer** and **the supervisor**, *probably nobody else*. (!)

Like many other degree projects in computer science there will be an implementation part. I will do my best to make sure that it does not swell uncontrollably or take up too much time.

The goal of this project is ***not*** to develop a ready-made tool for the client – Region Uppsala. It is much more likely to result in a *prototype* which hopefully can be further developed to become useful.