A Canvas course for degree projects

G. Q. Maguire Jr.

In order to use Canvas to keep track of students doing their degree projects, one of the first questions is how to get the students into this course. This question occurs because students who are not in a course cannot submit any assignments, hence in order to help automate degree project procedures we need to get the students into this course. One method of doing this is a course that students can self-enroll in. Here we will assume that the course has been made visible within the “Institution”.

An example of the process is shown below and inserted into the course: <https://kth.instructure.com/courses/751>.

The existing (former ICT) based KEX and MEX courses can be found at:

* <https://kth.instructure.com/courses/1585>
* <https://kth.instructure.com/courses/1586>

The slides from my presentation at the previous teacher meeting are at: <https://www.kth.se/social/files/5b7d561456be5bf8b3bb9f3f/Facilitating_Degree_projects-20180819a.pdf>

# Self-enrollment

Courses with self-enrollment are described in Canvas Doc Team, “How do I enable course self-enrollment with a join code or secret URL?” Last modified on 17 Nov. 2018 - <https://community.canvaslms.com/docs/DOC-13116-415274473>. The process is to go to the course Settings page, click on the link at the bottom saying “more options”, this will show a check box beside the string “Let students self-enroll by sharing with them a secret URL”. If you check this box and then you will get a check box and text that says “Add a "Join this Course" link to the course home page”. If you check this and the click the Update Course Details button – it will enable self-enrollment. If you look at the settings page you will have the URL that students can use to join this course, for example:

This course has enabled open enrollment. Students can self-enroll in the course once you share with them this URL: <https://kth.instructure.com/enroll/EG4TT9>

Now that students can self-enroll in the course the next step is to have an gatekeeper assignment (see [https://kth.instructure.com/courses/11/pages/followup-to-stortraffen-20181115](https://kth.instructure.com/courses/11/pages/followup-to-stortraffen-20181115?module_item_id=109349) ). An example of such an assignment is shown in Figure 1.

# Gatekeeper assignment to check for student meeting prerequisites

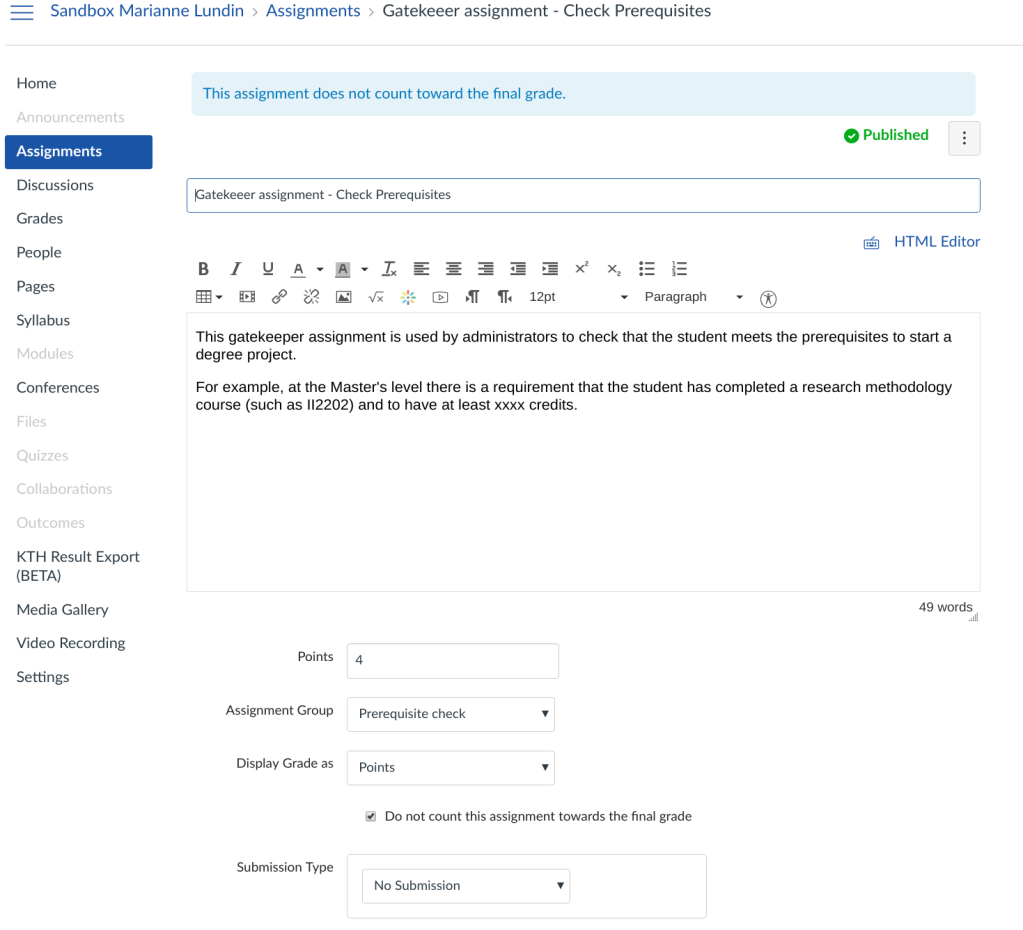


Figure : Gatekeeer assignment - Check Prerequisites

Next we add a protected module as shown in Figure 2. This protected module starts with a Quiz. This quiz (“Information om exjobbsprojekt/Information for degree project”) will get the relevant information from the student (as was previously supplied via the form **UT-EXAR (ANSÖKAN OM EXAMENSARBETE/*APPLICATION FOR DEGREE PROJECT)***), see

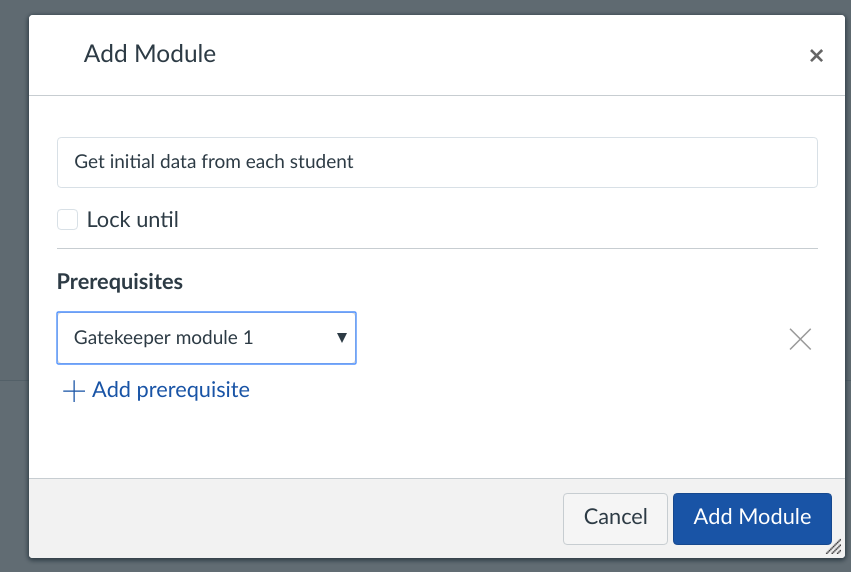


Figure : Module to get student details

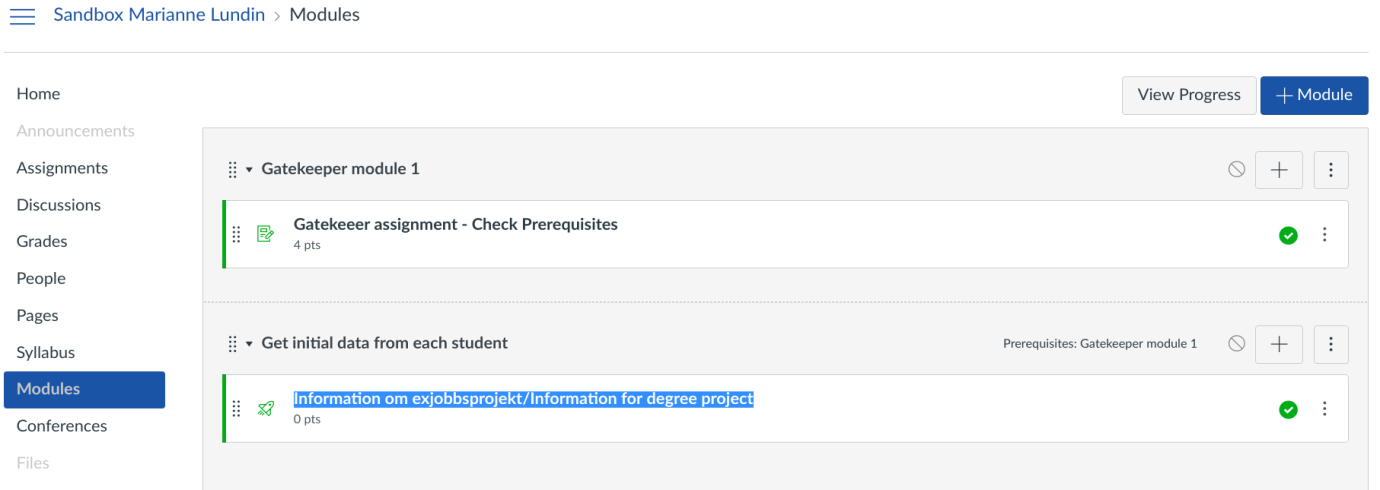


Figure : View of modules after importing the quiz

We adjust the gatekeeper module to require at least 3 points out of 4 as shown in Figure 4.

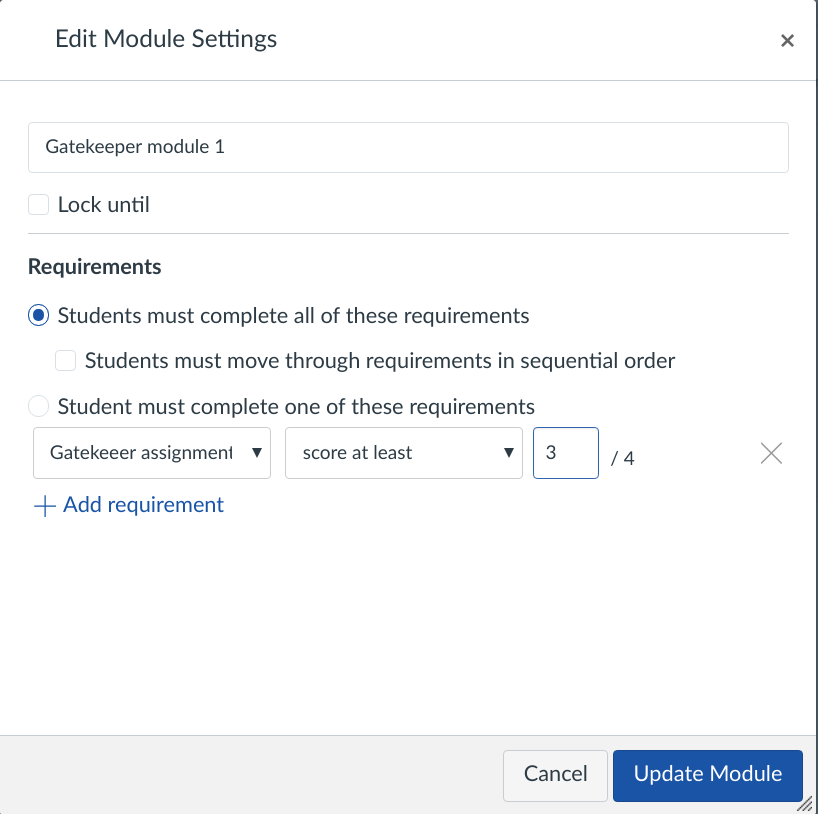


Figure : Setting requirements for the gatekeeper module

The view of the modules is as shown in Figure 5. Once an administrator has assigned 3 or more points for the student, then the student will have access to the next module (with the quiz in it).

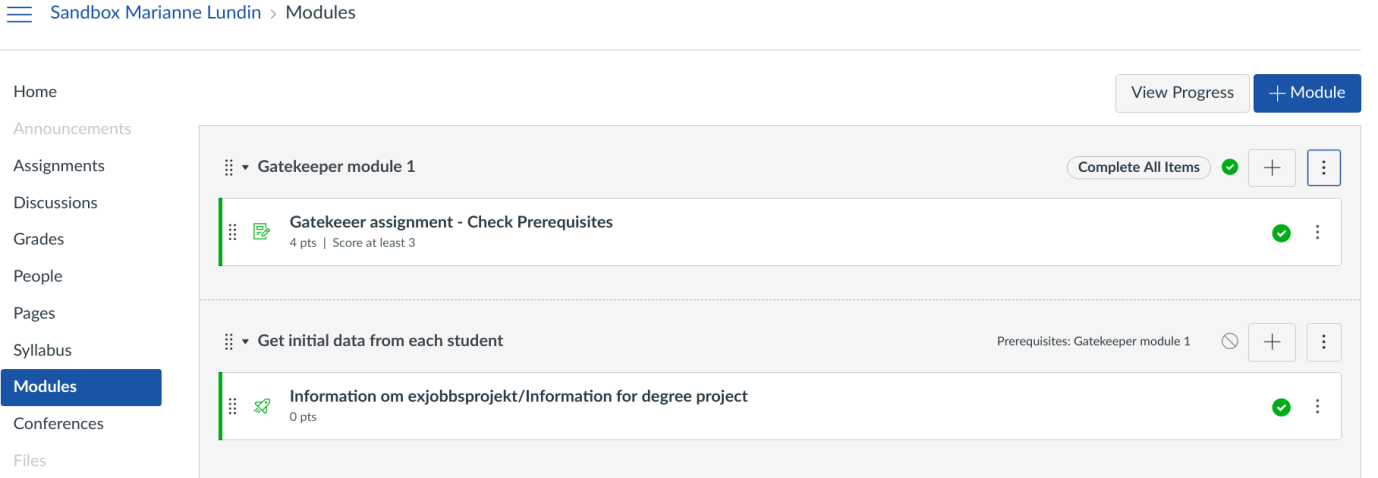


Figure : State of modules after setting the requirements for the gatekeeper

One method of encoding information into the points might be: 1=Research Methodology course completed, 2=requirement minimum number of points complete, 3 = both 1 and 2, and 4 could be used to enable the GA to grant an exception allowing a student to start without either of the requirements having been met.

# Getting preliminary data from the student

The quiz “Information om exjobbsprojekt/Information for degree project” is designed to collect the information needed before a student starts a degree project. The method is to use an ungrade survey as shown in Figure 6. The survey is available in both English and Swedish.

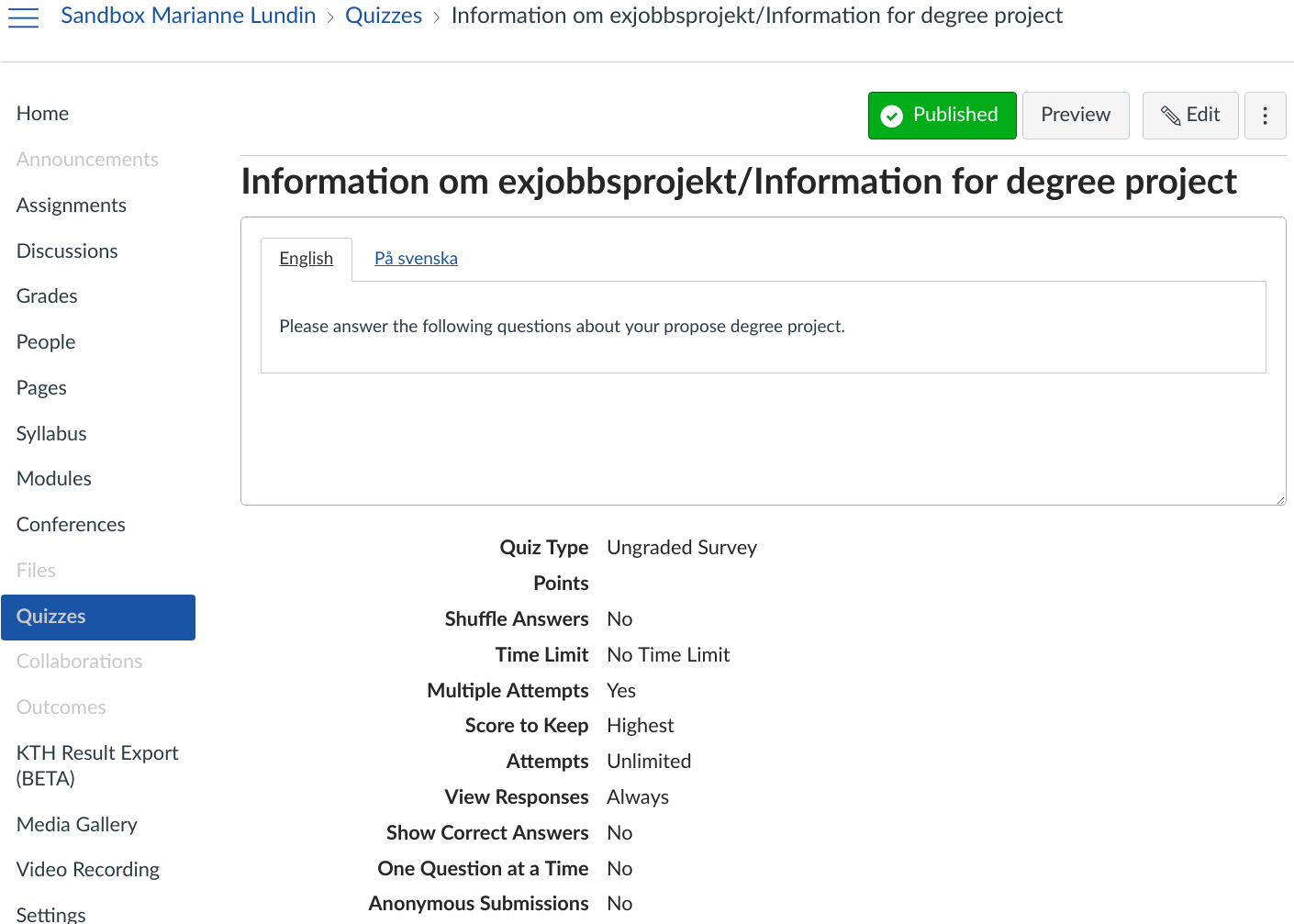


Figure : Survey quiz

Once the student has completed the quiz, a program can be used to extract the information and populate custom columns in the gradebook. (Assuming that the custom columns have already been created – as per Section xxx.

To get the survey submissions one first gets the list of submissions for the assignment with:

GET /api/v1/courses/:course\_id/assignments/:assignment\_id/submissions

Next you can ask for each student’s submission with:

GET /api/v1/courses/:course\_id/assignments/:assignment\_id/submissions/:user\_id

Note that you need to include the option: include[]=submission\_history

Note that the program remains to be written and would need to be run as a batch job or triggers by an LTI call connected to a button in the course. One can probably leverage the program get\_submission.py .

# Other assignments in the course

The course contains assignments for each of the things that a student needs to submit (proposal, alpha draft, beta draft, draft for opponent, final report, etc.) See the existing courses for how this can be done. NB: There should only be a single assignment for submitting the proposal and not one per program.

# Actually registering the student for the course

This step would initially be done manually by an administrator once the student’s proposal has been approved and an examiner assigned. The student should be added to a section for this examiner when the student is assigned to this examiner. This enables each examiner to easily see the status of the students that they are responsible for. See the existing KEX and MEX courses for examples of this.

Once an examiner is assigned a given student, then they will assign an academic adviser and the student should be added to a section for this academic adviser. This enables each academic adviser to easily see the status of the students that they are responsible for. See the existing KEX and MEX courses for examples of this.

# Creating custom columns in the gradebook

For each element of data that you want to record from the survey in the gradebook, you need to create a custom column. This can easily be done with a program, as follows:

insert-column.py course\_id column\_name

There is also a program that will allow you to put a column of data in from a spreadsheet:

insert-custom-columns-from-spreadsheet.py course\_id column\_name\_1 column\_name\_2 ...

# Automation of prerequisite checking

Given the existence of a read API to LADOK it should be possible to mechanically check if a student has completed the required methodology course (given a list of acceptable courses) and total number of points.

# Replace static quiz/survey with dynamic survey

Use an LTI plugin to call a program to do a dynamic survey or perhaps use via a Möbius quiz.

Given a means to get information about what program a student is in, the identification of which course code the student should register for can be automated – rather than having to be a question.

The list of possible examiner could come from KOPPS.

# Given a means to get information about what program a student is in

Given information about which program a student is in, the student could be placed into a section for student of this program. This will facilitate PAs being able to check the status of the students in their program. See the existing KEX and MEX courses for examples of this.

# Working notes

If one enables the course features (via the Settings->Features page): “Learning Mastery Gradebook” and “Student Learning Mastery Gradebook” (Note that I also enabled the “New Gradebook”.

# My answers (2019.01.15) to the appendix of a query(“Uppdrag examensarbeten och Canvas?”) from Ann Lantz.

> Bilaga 1. Prioriteringslista anpassning Canvas kurser för examensarbeten vid EECS

> 1. KTH:s blankett för anmälan till examensarbete.

> 2. Interaktiv självvärdering sk. bedömningsmallar för studenter, handledare och examinatorer.

> 3. Checklista för studenters aktiva deltagande i seminarier; muntlig presentation, opposition.

> 4. Quiz för insamling av data.

> 5. Möjliggöra presentation, listning samt sortering av student, kurs, handledare, examinator, uppdragsgivare, titel, abstract, rapport

> 6. Automatisera delar av eller alla punkter som ingår i: Studenten sänder slutversionen av den skriftliga rapporten (inklusive omslag) till examinator som kontrollerar att de ämnesmässiga uppgifterna stämmer. Examinator sänder den slutgiltiga PDF:en inklusive uppgift om kurskod, betyg och examinationsdatum till UK som förser rapporten med Trita-nummer, publicerar rapporten i Diva, lägger in betyg i Ladok, meddelar examinator när slutligt betyg ska göras.

> 7. Automatisk koppling till Events på EECS för annonsering av examensarbeten

> 8. Kursvärdering

> 9. Automatiskt larm och avisering 10 månader efter att examensarbetet startats.​

Reponses:

1. KTH:s blankett för anmälan till examensarbete.

I have a preliminary (static) version of this and reviewed it this afternoon (2019.01.15) with Marianne Lundin <marianlu@kth.se> and Bianca Lanciai <biancal@kth.se>. Bianca will go through it carefully and let me know if there are changes that they need.

One can see an example of the survey (quiz) at <https://kth.test.instructure.com/courses/12682> and a more complete degree project course at <https://kth.test.instructure.com/courses/12683>

Note that Question 6 in the quiz at https://kth.test.instructure.com/courses/12682 shows a more advanced version of the selection of an examiner than the version in <https://kth.test.instructure.com/courses/12683> . This version lists the eligible examiners for each course code as they are in KOPPS at the time I ran the program. I think that this version of the question is better than Question 5 which has a single list that it the union of all examiners.

2. Interaktiv självvärdering sk. bedömningsmallar för studenter, handledare och examinatorer.

This could probably be done via the Learning Mastery Gradebook for supervisor and examiner. It could be done via a survey for the student. Currently, I have little experience with the Learning Mastery Gradebook.

3. Checklista för studenters aktiva deltagande i seminarier; muntlig presentation, opposition.

The oral presentation is already an assignment in the course done for ICT (Presentationsseminarium/Presentation seminar - see the Canvas page <https://kth.instructure.com/courses/1586/assignments/4415> )

The written opposition is also already an assignment in the course done for ICT (Opponeringsrapport/Opposition report - see the Canvas page <https://kth.instructure.com/courses/1586/assignments/4283>

It would be easy to add two active participation assignments that do not require submission, but rather the examiner at the presentation would mark these as completed for the relevant student. [Added 2019.01.23]

4. Quiz för insamling av data.

See #1 for static survey (quiz) for UT-EXAR form.

The dynamic quiz is not done and I do not have an estimate of when the students are likely to be finished with it.

There is a quiz for project plan (Projekt Plan/Project plan Q - see the Canvas page <https://kth.instructure.com/courses/1586/quizzes/1431> )

5. Möjliggöra presentation, listning samt sortering av student, kurs, handledare, examinator, uppdragsgivare, titel, abstract, rapport

In the grade book for the course done for ICT (2nd cycle) - see https://kth.instructure.com/courses/1586/gradebook you can already sort by student name (A-Z or Z-A), by course code, examiner, supervisor, planned start date, tentative title, KTH unit, Company, External contact, Student approval (for DiVA publication of full text), DiVA-URN, and TRITA number.

The final title, abstract, and report itself would all be in DiVA and DiVA is searchable. I also have tools to export this data from DiVA in BibTeX form (https://github.com/gqmaguirejr and this works in both English and Swedish). Additionally, DiVA provides exports as spreadsheets, RSS feeds, etc.

6. Automatisera delar av eller alla punkter som ingår i:

Studenten sänder slutversionen av den skriftliga rapporten (inklusive omslag) till examinator som kontrollerar att de ämnesmässiga uppgifterna stämmer. Examinator sänder den slutgiltiga PDF:en inklusive uppgift om kurskod, betyg och examinationsdatum till UK som förser rapporten med Trita-nummer, publicerar rapporten i Diva, lägger in betyg i Ladok, meddelar examinator när slutligt betyg ska göras.

The examiner would not need to send the PDF, as this will already be in the Canvas course in the entry for the assignment (Examensarbete inlämnande/Final thesis submission - see the Canvas page https://kth.instructure.com/courses/1586/assignments/4281). When the examiner approve the thesis as complete, it should be ready for UK to process. The two students that I had earlier showed it is possible to extract the authors, title, abstracts, keywords, and other data from the PDF file. Their code generated a MODS file that can be imported into DiVA. However, there is no API to be able to automatically put the meta data and the thesis into DiVA. Ideally, it should be possible to automatically assign the TRITA number (and possibly a DOI or ISBN number) to the thesis and then generate the cover. The cover can automatically be applied to the PDF and produce a file that can be (manually) put into DiVA.

When there is integration between Canvas and Ladok, it should be possible for the Examiner to enter the final grade in Canvas and have this sync'd to Ladok. The examiner would still need to login into Ladok to certify the grade.

7. Automatisk koppling till Events på EECS för annonsering av examensarbeten

I currently generate these manually in both Canvas and Polopoly, but one could extract the data from the PDF for the opponent and then generate these two forms of announcements. For example, see https://kth.instructure.com/courses/1586/discussion\_topics/54827 and <https://www.kth.se/en/eecs/kalender/optimizing-the-performance-of-tls-libraries-for-sgx-1.867824?date=2019-01-07&orgdate=2019-01-07&length=1&orglength=1>

8. Kursvärdering

The LEQ can be done and posted on KTH Social and presumably the new Course Catelog. One small problem is that of course there is basically no ending time for all the students in the course, as degree projects are completed at different times all through the year.

I have no idea of what the exjobb coordinators do about course evaluations.

9. Automatiskt larm och avisering 10 månader efter att examensarbetet startats.​

Assuming that the planned start date is correct or perhaps using the time when the project plan is approved, one could have an application that generated a warning message for the student at 10 months (or another time point).

The code for what I have done is at <https://gitr.sys.kth.se/maguire/E-learning>

# My e-mail to Marianne Lundin on 2019.01.30 about the importance of working with the IT unit

Although replacing the UT-EXAR form is coupled to someone's priorities #1 and #4, the alternatives are:

1. Use something like the existing static form that I have already made
2. Be smarter with a dynamic survey and avoid asking unnecessary questions.

Currently A is basically done (just awaiting feedback from others for (hopefully) small changes). In contrast, the two students working on a degree project to realize alternative B, will have to fake that the program information is known - since without the help of KTH's IT unit - I cannot automatically get information about what program a student is in. Even though this information is in Ladok and UG - I have no means to get it. As you can see from the forwarded message below, I ask about getting this information already in November and have not heard anything about it.

If I could know what program a student is in, then in many cases I can:

1. avoid asking the student if they want a P/F or A-F grade
2. automatically form sections for each program in the Canvas course - so that a PA can easily see all of the students in his or her program (by simply viewing that section). I manually did the assignments of students by program to sections for the two ICT degree project Canvas course rooms)
3. potentially reduce the list of alternative courses to a small number (in the ideal case to 1 - hence I can avoiding even asking about the course code!)

There are some postings in the Canvas Community about the usefulness of knowing which program a student is in:

* This posting in the Canvas LMS Community also looks at using custom data to store the student's major (this is what the program of student is called in US universities).

See https://community.canvaslms.com/thread/5951

Specifically the posting by Kenneth MayerNov 22, 2017 5:31 PM

* Another usage is reporting data for cohorts (such as by school, major/program, ...), see https://community.canvaslms.com/thread/22750-is-there-a-way-to-include-metadata-for-people-in-canvas-for-outcome-reporting

I am concerned that it is soon going to be February and this means of the 2nd cycle students will already have started their degree projects for this year and hence the effort will not have any impact for this year for these students. While there is some hope that perhaps something could be done for 1st cycle students who only start in period 4.

## Status

If A above is acceptable, then I have a large part of priorities 1 & 4 done. To go further with #1 would involve doing the eligibility check of whether students can start their degree project by check if they have the requisite number of points (would require using the Ladok API) and if they have take an approved research methodology course (I currently do not know where I can get this information automatically - but might be able to mine this from KOPPS data).

* With regard to #3, while it is not a check list - there are assignments for each of these in what I have already done - so it is easy to both the student and the examiner to see what needs to be done and has been done
* With regard to #5, as noted previously Canvas already lets you sort and view the students by these factors (except for title, abstract, and report - but these functions are already available via DiVA and in the Canvas gradebook there is a place for the working title and a place for the URL to the DiVA entry).
* Parts of #6 have been shown and I hope to explore one way of doing these and #7 via a new pair of 1st cycle students. [I had hoped to meet with them this afternoon, but they cannot meet until tomorrow afternoon.]
* I will need further input from others about what they think #8 includes.
* #9, might be as simple as putting an entry in the student's Canvas calendar for 10 months after they have started and then removing it if the thesis is completed. [I also think that this could be used to generate calendar events for milestones along the way - but I have not explored this (yet).]
* I have been working on some ideas for how to do #2. Trying to make these ideas work will be something that I have to work on today. The largest impediments are that I have little experience with Canvas rubrics and Canvas does not let a student be assigned as their own peer reviewer (however, I may have a means of working around this but it will take some effort).

## What are the costs of the delay?

1. Since the former CSC does not even use the UT-EXAR form - this means at a minimum 10% of Bianca's time goes to collecting the data for the questions of whether the student gives their permission for the full text of the thesis to be made available via DiVA.
2. An unknown number of paper forms will go astray and this will have high costs later (when extra effort has to be made or when we lose the monies for these students)
3. Manual processing will continue at a higher cost than if automated (sadly you will probably hire the extra person that you mentioned)

## Knowing a student’s program

There is a need in a number of settings to know what program of study a student is in.

Examples of usage:

1. Creating sections within a course based upon the program of study
2. for degree projects

For example, in the 2nd cycle degree project course <https://kth.instructure.com/courses/1586> and in the 1st cycle degree project course - <https://kth.instructure.com/courses/1585> there is a section for each of the programs, so that the PA can easily see the status of all of the students in their programs with respect to completing their degree projects.

1. for a course with participants from many programs and specific instructors for these students

For example, in II2202 - <https://kth.instructure.com/courses/6434> I have created sections for each of the instructors, but also have sections for students from specific programs and year - such as Henrik's section for DASC yr1

1. For giving PAs access to the gradebook in courses - but limiting them to seeing only the students in their program

XXX has been developing a new set of roles for use at KTH, including roles for PA, GA, Prefekt, etc.

If students were in a program specific section, it would be possible to give a PA access to a course, but limit them to seeing only the students in their program.

1. To provide this data to programs involved by LTI

For example, the planned dynamic survey to replace the UT-EXAR form.

### Technical implementation

The information about which program of study a student is in resides in UG (accessible via LDAP) and in Ladok.

Suggestion:

This date could be populated nightly when users are sync'd to Canvas.

Add this information to a user's custom data (see https://canvas.instructure.com/doc/api/users.html#method.custom\_data.set\_data)

PUT /api/v1/users/:user\_id/custom\_data(/\*scope)

Scope: url: PUT /api/v1/users/:user\_id/custom\_data(/\*scope)

### Examples

curl 'https://kth.instructure.com/api/v1/users/<user\_id>/custom\_data/program\_of\_study \

-H 'Content-Type: application/json' \

-X PUT \

-d '{

"ns": "se.kth.canvas-app.program\_of\_study",

"data": {

"programs": [{"code": "TIVNM-DASC", "name": "ICT Innovation, (TIVNM), Data Science (DASC) Program", "start": 2016}]

}

}' \

-H 'Authorization: Bearer <token>'

You can use "self" as the <user\_id>:

curl 'https://kth.test.instructure.com/api/v1/users/self/custom\_data' \

-H 'Content-Type: application/json' \

-X PUT \

-d '{

"ns": "se.kth.canvas-app.program\_of\_study",

"data": {

"programs": [{"code": "TIVNM-DASC", "name": "ICT Innovation, (TIVNM), Data Science (DASC) Program", "start": 2016}]

}

}' \

-H 'Authorization: Bearer <token>'

==>

{"data":{"programs":[{"code":"TIVNM-DASC","name":"ICT Innovation, (TIVNM), Data Science (DASC) Program","start":2016}]}}

You can also get the stored value:

curl 'https://kth.test.instructure.com/api/v1/users/self/custom\_data' \

-X GET \

-F 'ns=se.kth.canvas-app.program\_of\_study' \

-H 'Authorization: Bearer <token>'

==>

{"data":{"programs":[{"code":"TIVNM-DASC","name":"ICT Innovation, (TIVNM), Data Science (DASC) Program","start":2016}]}}

If you do not supply the name space:

curl 'https://kth.test.instructure.com/api/v1/users/self/custom\_data' \

-X GET \

-H 'Authorization: Bearer <token>'

==>

{"message":"invalid namespace"}

If you do not supply the complete name space (subspace):

curl 'https://kth.test.instructure.com/api/v1/users/self/custom\_data' \

-X GET \

-F 'ns=se.kth.canvas-app' \

-H 'Authorization: Bearer <token>'

==>

{"message":"no data for scope"}

One can even have multiple programs:

curl 'https://kth.test.instructure.com/api/v1/users/self/custom\_data' \

-H 'Content-Type: application/json' \

-X PUT \

-d '{

"ns": "se.kth.canvas-app.program\_of\_study",

"data": {

"programs": [{"code": "TIVNM-DASC", "name": "ICT Innovation, (TIVNM), Data Science (DASC) Program", "start": 2016},

{"code": "TSEDM", "name": "Software Engineering of Distributed Systems (TSEDM) Program", "start": 2018}

]

}

}' \

-H 'Authorization: Bearer <token>'

==>

{"data":{"programs":[{"code":"TIVNM-DASC","name":"ICT Innovation, (TIVNM), Data Science (DASC) Program","start":2016},{"code":"TSEDM","name":"Software Engineering of Distributed Systems (TSEDM) Program","start":2018}]}}

# Difficulty in listing potential degree courses per program

Degree courses per program

programs\_in\_the\_school

['CDATE', 'CELTE', 'CINTE', 'CMETE', 'TCOMK', 'TCOMM', 'TCSCM', 'TEBSM', 'TEFRM', 'TELPM', 'TIDAB', 'TIEDB', 'TIELA', 'TIETM', 'TIMTM', 'TINNM', 'TIVNM', 'TMAIM', 'TMMTM', 'TNTEM', 'TSCRM', 'TSEDM']

Based upon getting the list of degree projects for each program, based upon the syllabi in KOPPS:

degree\_project\_code\_per\_program(programs\_in\_the\_school)

Program CDATE

course codes: {'DD143X', 'DD142X'}

Program CELTE

course codes: {'EJ111X', 'EP111X', 'EK111X', 'EN111X', 'EH111X', 'EF111X', 'EQ111X', 'EI111X', 'EL111X', 'ED111X', 'EF112X', 'EG111X'}

Program CINTE

course codes: {'II143X'}

Program CMETE

course codes: {'DM128X', 'DM129X'}

Program TCOMK

course codes: {'II143X'}

Program TCOMM

course codes: {'IK251X', 'IL246X', 'IK223X'}

Program TCSCM

course codes: {'DA222X', 'DD221X', 'DA221X', 'DA231X', 'DA225X', 'DA224X'}

Program TEBSM

course codes: set() --- There are not any degree project courses in the syllabi

Program TEFRM

course codes: set() --- There are not any degree project courses in the syllabi

Program TELPM

course codes: set() --- There are not any degree project courses in the syllabi

Program TIDAB

course codes: {'II121X', 'IK150X', 'IK121X', 'IV101X', 'II142X'}

Program TIEDB

course codes: {'IL142X'}

Program TIELA

course codes: {'HE110X', 'HE111X'}

Program TIETM

course codes: {'MJ273X'}

Program TIMTM

course codes: {'DA232X', 'DA222X', 'DA221X'}

Program TINNM

course codes: set() --- There are not any degree project courses in the syllabi

Program TIVNM

course codes: {'IL222X'}

Program TMAIM

course codes: {'DA221X', 'DA233X', 'SF288X', 'SF299X', 'DA222X', 'DD221X', 'DA225X', 'DA224X'}

Program TMMTM

course codes: {'DA221X', 'DM228X', 'DA225X', 'DA234X', 'DA224X'}

Program TNTEM

course codes: {'IF246X'}

Program TSCRM

course codes: set() --- There are not any degree project courses in the syllabi

Program TSEDM

course codes: {'II246X'}

By looking at the target group ('Målgrupp') for each of the relevant degree project courses:

course code: DA221X has target group= None

course code: DA222X has target group= ['Endast öppen för studenter antagna på civilingenjörsprogrammet i Datateknik (CDATE) eller civilingenjörsprogrammet i Medieteknik (CMETE). Studenterna måste även vara antagna på ett av CSC-skolans masterprogram.']

course code: DA223X has target group= ['Öppen för alla studenter förutom studenter antanga på något av CSC-skolans utbildningsprogram.']

course code: DA224X has target group= None

course code: DA225X has target group= None

course code: DA226X has target group= ['Öppen för alla förutom studenter antanga på något av CSC-skolans utbildningsprogram.']

course code: DA231X has target group= None

course code: DA232X has target group= ['Endast öppet för studenter antagna på Masterprogrammet i Interaktiv Medieteknik']

course code: DA233X has target group= ['Endast öppet för studenter antagna på masterprogrammet i Maskinilärning.']

course code: DA234X has target group= ['Endast öppet för studenter antagna på masterprogrammet i Media Management']

course code: DA235X has target group= ['Endast för studenter antagna på civilingenjörsprogrammet Industriell ekonomi']

course code: DA236X has target group= None

course code: DA239X has target group= None

course code: ED222X has target group= None

course code: ED225X has target group= ['Studenter från\xa0KTH programmen i Elektroteknik eller\xa0Teknisk fysik\xa0eller liknande', 'nationell eller internationell', 'bakgrund.']

course code: ED226X has target group= ['Studenter från KTH programmen i Elektroteknik eller Teknisk fysik eller liknande', 'nationell eller internationell', 'bakgrund.']

course code: EF225X has target group= None

course code: EF226X has target group= ['Öppen för alla program.']

course code: EF227X has target group= None

course code: EF231X has target group= None

course code: EF232X has target group= None

course code: EF233X has target group= ['Öppen för alla program.']

course code: EG201X has target group= None

course code: EG230X has target group= None

course code: EH231X has target group= None

course code: EH241X has target group= None

course code: EH251X has target group= None

course code: EH252X has target group= None

course code: EH253X has target group= None

course code: EH257X has target group= None

course code: EH258X has target group= None

course code: EH259X has target group= None

course code: EI252X has target group= None

course code: EI253X has target group= None

course code: EI254X has target group= None

course code: EI255X has target group= None

course code: EI270X has target group= None

course code: EJ210X has target group= None

course code: EJ211X has target group= None

course code: EJ212X has target group= None

course code: EK211X has target group= None

course code: EK212X has target group= ['Studerande på Civilingenjörsprogrammen i Elektroteknik', 'Teknisk fysik', 'Medicinsk teknik eller motsvarande samt på de Masterprogram som är knutna till dessa utbildningar.']

course code: EK213X has target group= ['Studerande på Civilingenjörsprogrammen i Elektroteknik', 'Teknisk fysik', 'Medicinsk teknik eller motsvarande samt på de Masterprogram som är knutna till dessa utbildningar.']

course code: EL201X has target group= ['Öppen för alla studenter', 'notera dock förkunskaper.']

course code: EL205X has target group= None

course code: EP240X has target group= None

course code: EP241X has target group= None

course code: EP242X has target group= None

course code: EP243X has target group= None

course code: EP247X has target group= None

course code: EP248X has target group= None

course code: EP249X has target group= None

course code: EQ272X has target group= ['Öppen för alla master studenter. Individuell antagning av examinator']

course code: EQ274X has target group= ['Öppen för alla master studenter. Individuell antagning av examinator.']

course code: EQ275X has target group= ['Öppen för alla master studenter. Individuell antagning av examinator.']

course code: EQ276X has target group= ['Öppen för alla master studenter. Individuell antagning av examinator.']

course code: IF225X has target group= None

course code: IF226X has target group= ['TNTEM']

course code: IF227X has target group= None

course code: IF245X has target group= ['CMIEL']

course code: IF246X has target group= ['TNTEM']

course code: IF247X has target group= None

course code: IF249X has target group= None

course code: II225X has target group= None

course code: II226X has target group= ['TSEDM', 'TDISM', 'TIVNM']

course code: II227X has target group= None

course code: II245X has target group= ['CINTE']

course code: II246X has target group= ['TSEDM', 'TDISM', 'TIVNM']

course code: II247X has target group= None

course code: II249X has target group= None

course code: IL226X has target group= ['TCOMM', 'TIVNM', 'TEBSM']

course code: IL227X has target group= None

course code: IL228X has target group= None

course code: IL246X has target group= ['TCOMM', 'TIVNM', 'TEBSM']

course code: IL247X has target group= None

course code: IL248X has target group= ['CINTE']

course code: IL249X has target group= None

course code: IT225X has target group= None

course code: IT245X has target group= ['CMIEL']

In conclusion, it is difficult to know exact which courses are really possible in a given degree program - absent a matrix such as show for the former ICT programs that I have worked with - as shown at https://people.kth.se/~maguire/

# Creating fake users in a VM version of Canvas for testing

As of 2019.01.30 I created a program create-fake-users-in-course.py to create fake users in a Canvas instance and then enroll these users as students into a Canvas course. You can find the source code at <https://github.com/gqmaguirejr/Canvas-tools>.

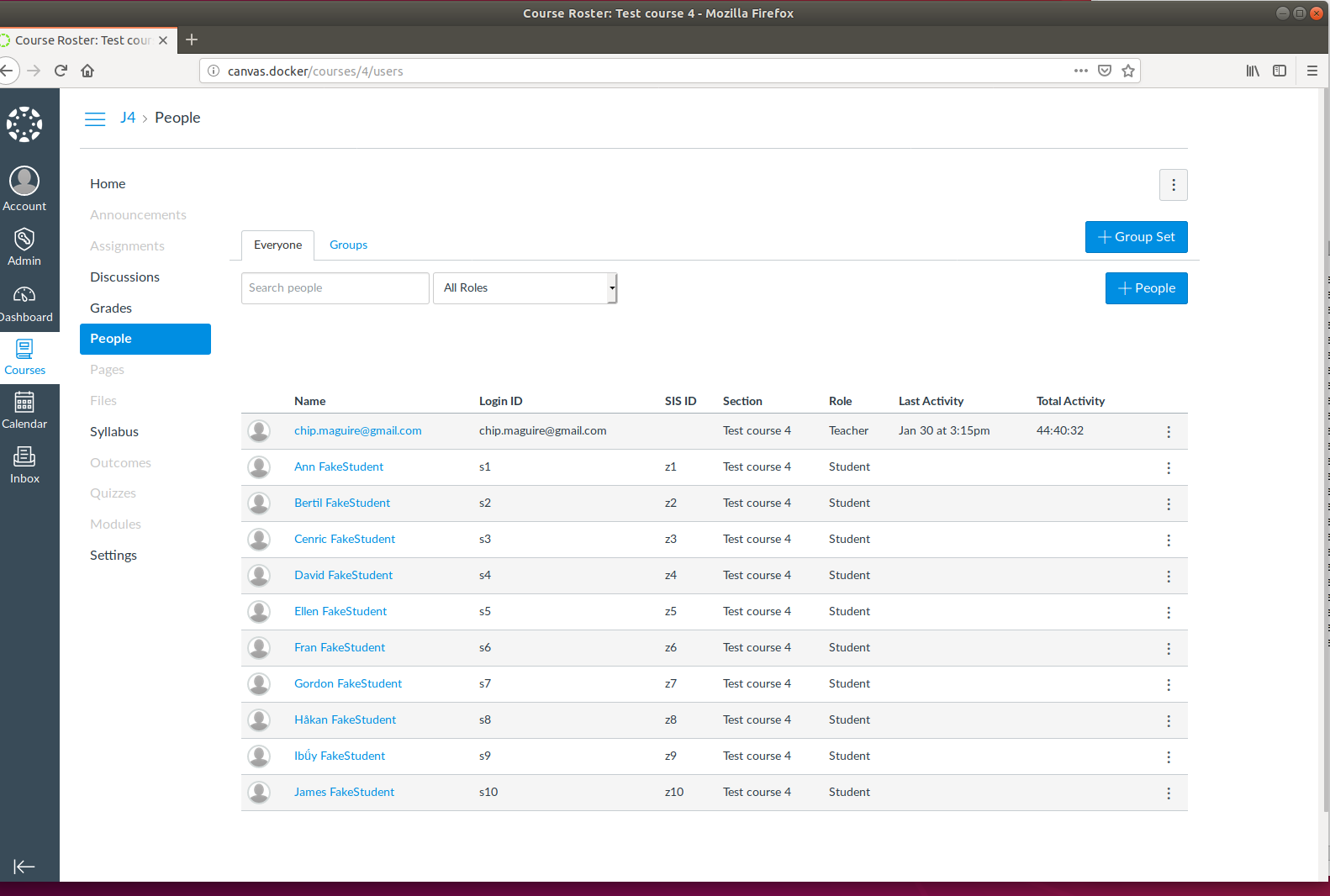


Figure : Examples of fake users that were created by the program

Note that these users are also augmented with information about the programs of study that they are in (note that all of the data is fake and built into the program’s source code – however, the names of the programs are actually correct and take from KOPPS).

result of setting custom data for user Ann FakeStudent is {'data': {'programs': [{'code': 'CINTE', 'name': 'Degree Programme in Information and Communication Technology', 'start': 2016}]}}

result of getting custom data for user Ann FakeStudent is {'data': {'programs': [{'code': 'CINTE', 'name': 'Degree Programme in Information and Communication Technology', 'start': 2016}]}}

result of setting custom data for user Bertil FakeStudent is {'data': {'programs': [{'code': 'CDATE', 'name': 'Degree Programme in Computer Science and Engineering', 'start': 2016}]}}

result of getting custom data for user Bertil FakeStudent is {'data': {'programs': [{'code': 'CDATE', 'name': 'Degree Programme in Computer Science and Engineering', 'start': 2016}]}}

result of setting custom data for user Cenric FakeStudent is {'data': {'programs': [{'code': 'CINTE', 'name': 'Degree Programme in Information and Communication Technology', 'start': 2016}]}}

result of getting custom data for user Cenric FakeStudent is {'data': {'programs': [{'code': 'CINTE', 'name': 'Degree Programme in Information and Communication Technology', 'start': 2016}]}}

result of setting custom data for user David FakeStudent is {'data': {'programs': [{'code': 'CINTE', 'name': 'Degree Programme in Information and Communication Technology', 'start': 2016}]}}

result of getting custom data for user David FakeStudent is {'data': {'programs': [{'code': 'CINTE', 'name': 'Degree Programme in Information and Communication Technology', 'start': 2016}]}}

result of setting custom data for user Ellen FakeStudent is {'data': {'programs': [{'code': 'CDATE', 'name': 'Degree Programme in Computer Science and Engineering', 'start': 2016}]}}

result of getting custom data for user Ellen FakeStudent is {'data': {'programs': [{'code': 'CDATE', 'name': 'Degree Programme in Computer Science and Engineering', 'start': 2016}]}}

result of setting custom data for user Fran FakeStudent is {'data': {'programs': [{'code': 'TEBSM', 'name': "Master's Programme, Embedded Systems, 120 credits", 'start': 2016}]}}

result of getting custom data for user Fran FakeStudent is {'data': {'programs': [{'code': 'TEBSM', 'name': "Master's Programme, Embedded Systems, 120 credits", 'start': 2016}]}}

result of setting custom data for user Gordon FakeStudent is {'data': {'programs': [{'code': 'CDATE', 'name': 'Degree Programme in Computer Science and Engineering', 'start': 2016}]}}

result of getting custom data for user Gordon FakeStudent is {'data': {'programs': [{'code': 'CDATE', 'name': 'Degree Programme in Computer Science and Engineering', 'start': 2016}]}}

result of setting custom data for user Håkan FakeStudent is {'data': {'programs': [{'code': 'TCOMM', 'name': "Master's Programme, Communication Systems, 120 credits", 'start': 2016}]}}

result of getting custom data for user Håkan FakeStudent is {'data': {'programs': [{'code': 'TCOMM', 'name': "Master's Programme, Communication Systems, 120 credits", 'start': 2016}]}}

result of setting custom data for user Ibǘy FakeStudent is {'data': {'programs': [{'code': 'CELTE', 'name': 'Degree Programme in Electrical Engineering', 'start': 2016}]}}

result of getting custom data for user Ibǘy FakeStudent is {'data': {'programs': [{'code': 'CELTE', 'name': 'Degree Programme in Electrical Engineering', 'start': 2016}]}}

result of setting custom data for user James FakeStudent is {'data': {'programs': [{'code': 'TIVNM', 'name': "Master's Programme, ICT Innovation, 120 credits", 'start': 2016}]}}

result of getting custom data for user James FakeStudent is {'data': {'programs': [{'code': 'TIVNM', 'name': "Master's Programme, ICT Innovation, 120 credits", 'start': 2016}]}}

# KTH Canvas configuration files

On 26 November 2018, xxx told me:

If all you want is the look and feel of the production Canvas, but no data, you should be able to download one css file and one js file from Canvas and import it into your Canvas.

You can find them under https://kth.instructure.com/accounts/1/theme\_editor, if you click in "Upload" and then "show file". These can be uploaded in the same page in the other Canvas instance.

I did the above and found the CSS and Javascript files for the KTH Canvas instance uses the CSS file:   
 <https://instructure-uploads-eu.s3.eu-west-1.amazonaws.com/account_87790000000000001/attachments/1324683/no_flicker_styling.css>  
and the custom Javascript file:   
<https://instructure-uploads-eu.s3-eu-west-1.amazonaws.com/account_87790000000000001/attachments/149106/prism.js>

However, in the Canvas build in containers – there is no option to upload CSS files,as this option is not enabled[[1]](#footnote-1). Moreover, this version does not have S3 support built into it. The result is that there is no “Upload” tab at the top of the left menu (see Figure 8 versus Figure 9).

|  |
| --- |
|  |

Figure : Theme editor view in VM version of Canvas

|  |
| --- |
|  |

Figure : Theme editor as seen in the production KTH version of Canvas

Looking at the file “./lib/brandable\_css.rb” one can see the following lines:

def public\_brandable\_css\_folder

Pathname.new('public/dist/brandable\_css')

end

def default\_brand\_folder

public\_brandable\_css\_folder.join('default')

end

def default\_brand\_file(type, high\_contrast=false)

default\_brand\_folder.join("variables#{high\_contrast ? '-high\_contrast' : ''}-#{default\_variables\_md5}.#{type}")

end

This reveals that the location for the CSS file is 'public/dist/brandable\_css/default’ or for the high contrast version 'public/dist/brandable\_css/default-high\_contrast’.

The file ‘config/brandable\_css.yml’ has a line:

brandable\_variables\_json: app/stylesheets/brandable\_variables.json

The file ‘app/stylesheets/brandable\_variables.json’ defines the default configuration information for what colors are used for what, where the Canvas log is for branding ‘images/canvas\_logomark\_only@2x.png’ (which is actally under the public directory, i.e., it is at ‘public/images/canvas\_logomark\_only@2x.png’ and another for the variable "ic-brand-Login-logo" – which points to the image

# Adding functionality for publishing an announcement of a student’s oral presentation

The idea is that when the examine decides that a given student’s oral presentation should be schedule, the examiner is typically in the gradebook (having just looked at the beta draft of the thesis). This means that the examiner has a view such as shown in Figure 10.

|  |
| --- |
|  |

Figure : A view of a gradebook in Course 5

The examiner should be able to select the student from the left hand column of student names and then click to get to a page for this student, such as shown in Figure 11.

|  |
| --- |
|  |

Figure : A student as seen after clicking on their name in the gradebook.

Ideally, there should be a button to schedule the oral presentation. Here that button is labeled “TestButtonLeft”. The next section discusses the next step in realizing this idea.

# Calling an external tool with the URL of the page where the call comes from

I initially experimented with using an external tool added as a course\_navagation external tool (so it is added to the menu on the left as shown in Figure 12).

|  |
| --- |
|  |

Figure : Example of a view of a student's "People" page within a course

Unfortunately, I was not able to find any custom variables that could be used to pass the target user’s ID to the LTI tool or any means to pass the URL of the page where the button is. However, investigating the page with Firefox’s Inspector reveals the structure shown in Figure 12.

The inspector reveals that in the body of the page, there is a

<div id="application" class="ic-app">

inside this there is a

<div id="wrapper" class="ic-Layout-wrapper">

there is a

<div id="left-side" class="ic-app-course-menu list-view" style="display: block">

inside this there is:

<nav role="navigation" aria-label="Courses Navigation Menu">

<ul id="section-tabs">

<li class="section"><a href="/courses/5" title="Home" class="home" tabindex="0">Home</a></li>

<li class="section section-tab-hidden"><a href="/courses/5/announcements" title="Announcements" class="announcements" tabindex="0">Announcements<span id="inactive\_nav\_link" class="screenreader-only">\* No content has been added</span></a></li>

<li class="section"><a href="/courses/5/assignments" title="Assignments" class="assignments" tabindex="0">Assignments</a></li>

<li class="section"><a href="/courses/5/discussion\_topics" title="Discussions" class="discussions" tabindex="0">Discussions</a></li>

<li class="section"><a href="/courses/5/grades" title="Grades" class="grades" tabindex="0">Grades</a></li>

<li class="section"><a href="/courses/5/users" title="People" class="people active" tabindex="0">People</a></li>

<li class="section"><a href="/courses/5/wiki" title="Pages" class="pages" tabindex="0">Pages</a></li>

<li class="section section-tab-hidden"><a href="/courses/5/files" title="Files" class="files" tabindex="0">Files<span id="inactive\_nav\_link" class="screenreader-only">\* No content has been added</span></a></li>

<li class="section"><a href="/courses/5/assignments/syllabus" title="Syllabus" class="syllabus" tabindex="0">Syllabus</a></li>

<li class="section section-tab-hidden"><a href="/courses/5/outcomes" title="Outcomes" class="outcomes" tabindex="0">Outcomes<span id="inactive\_nav\_link" class="screenreader-only">\* No content has been added</span></a></li>

<li class="section section-tab-hidden"><a href="/courses/5/quizzes" title="Quizzes" class="quizzes" tabindex="0">Quizzes<span id="inactive\_nav\_link" class="screenreader-only">\* No content has been added</span></a></li>

<li class="section"><a href="/courses/5/modules" title="Modules" class="modules" tabindex="0">Modules</a></li>

<li class="section"><a href="/courses/5/external\_tools/3?display=borderless" title="TestButtonLeft" class="context\_external\_tool\_3" target="\_blank" tabindex="0">TestButtonLeft</a></li>

<li class="section"><a href="/courses/5/settings" title="Settings" class="settings" tabindex="0">Settings</a></li>

</ul>

</nav>

Within this we want to replace:

<li class="section"><a href="/courses/5/external\_tools/3?display=borderless" title="TestButtonLeft" class="context\_external\_tool\_3" target="\_blank" tabindex="0">TestButtonLeft</a></li>

with:

<li class="section"><a href="http://localhost:3597/announce" onclick="location.href=this.href+'?custom\_loc='+window.location.href;return false;"

title="TestButtonLeft" class="context\_external\_tool\_3" target="\_blank" tabindex="0">TestButtonLeft</a></li>

The magic here is to add the onclick action to add the value of the variable window.location.href as a value for the key "custom\_loc". Moreover, the external tool's URL which was "/courses/5/external\_tools/3?display=borderless" (i.e., a Canvas URL) to invoke the external tool is replaced with "http://localhost:3597/announce". This is designed to be used with an external server that processes a GET "/announce" request.

Now if one clicks the button "TestButtonLeft" - it dispatches to my Sinatra server (running on the local host at the indicate port) via the GET route /announce

Where it outputs:

in route for /announce

custom\_loc is http://canvas.docker/courses/5/users/5

request['custom\_loc'] is http://canvas.docker/courses/5/users/5

request.path\_info is /announce

request.path is /announce

params are {"custom\_loc"=>"http://canvas.docker/courses/5/users/5"}

The route itself begins:

get '/announce' do

puts("in route for /announce")

custom\_loc=params['custom\_loc']

puts("custom\_loc is #{custom\_loc}")

@str1=request['custom\_loc']

puts("request['custom\_loc'] is #{@str1}")

path\_info=request.path\_info

puts("request.path\_info is #{path\_info}")

path=request.path

puts("request.path is #{path}")

puts "params are #{params}"

This appears to be an ugly hack and I do not know how to automate this, but that will be the next step - now that I have shown that it is possible to pass the current page's URL to the external server. It certainly would have been easier if the current page's URL was something that one could get as an LTI custom parameter. Perhaps this could be: com.instructure.page.url or something similar.

For some related discussion see <https://community.canvaslms.com/message/136464-can-an-lti-call-pass-the-page-it-is-called-from>.

# First step towards being able to generate announcements

I have written a simple ruby program announce-presentation.rb that utilizes a Sinatra server to enable the user to make an announcement.

Consider the case when Ellen has submitted a draft for opponent. In this case the examiner has assigned James as Ellen’s opponent (as shown in Figure 13).

|  |
| --- |
|  |

Figure : James has been assigned as the peer reviewer for Ellen

Once the examiner has approved this draft (shown by the checkmark in Figure 14) the examiner goes to Ellen‘s page (under “People”) (see Figure 15) and then copies the URL for this page (as shown in ).

|  |
| --- |
|  |

Figure : Gradebook when Ellen has submitted a draft for her opponent

|  |
| --- |
|  |

Figure : The People page for Ellen

|  |
| --- |
|  |

Figure Select and copy the URL for Ellen's page

Next the examiner clicks on the button “TestButtonLeft” and the user will now be prompted to enter this URL as shown in Figure 16.

|  |
| --- |
|  |

Next the examiner clicks on “Submit” and will be presented with the screen shown in Figure 17.

|  |
| --- |
|  |

Figure : Pick the date for Ellen's oral presentation

After submitting the data, the program extracts information about the author, opponents, and the thesis and then presents this information to the examiner for approval (as shown in Figure 18).

|  |
| --- |
|  |

Figure : Announcment with information about opponent and thesis

After modifying this data the examiner clicks on the Approve” button and will see the screen shown in Figure 19.

|  |
| --- |
|  |

Figure : Approved announcement (just shown as parameters of the post)

Currently, much of the functionality is very limited (since it does not actually parse the submitted draft of the thesis that is for the opponent). The parsing of the thesis is just faked at this point with fixed values being returned. This will be corrected once the parsing of the PDF document is completed.

# Second step in generating an announcement

Some additional features were added to the code for generating announcements. We use the same student and opponent as in the previous section. Starting with the student’s People page and selecting the text of the URL (as shown in Figure 20).

|  |
| --- |
|  |

Figure : Ellen's people page with the URL selected and copied

Next the examiner clocks on the “TestButtonLeft” and enters the user’s URL from the previous page (see Figure 21).

|  |
| --- |
|  |

Figure : After pushing the button and entering the URL

Next the examiner enters the date, time, and location (as shown in Figure 22).

|  |
| --- |
|  |

Figure : After entering data and time and choosing a location

The examiner now gets to approve the information (and possibly modify it) then clicks Approve (see Figure 23).

|  |
| --- |
|  |

Figure : Information for the announcement for the examiner to approve

After approving the contents the examiner will see a screen with a confirmation (see Figure 24). Before this page is shown an announcement was inserted in to the Canvas course room (see Figure 25 and Figure 26).

|  |
| --- |
|  |

Figure : Confirmation page

|  |
| --- |
|  |

Figure : Announcements page

|  |
| --- |
|  |

Figure : The final announcement

# Polopoly calendar event

An example of a calendar event when view with the inspect in Chrome is:

|  |
| --- |
| <article id="mainContent" role="main" class="article standard calendarevent" vocab="http://schema.org/" typeof="Event" data-cid="1.867824" lang="en-UK">  <div class="preArticleParagraphs">  <h1 property="name">Optimizing the performance of TLS libraries for SGX</h1>  <div class="lead">        <p>Master's thesis presentation</p>      </div>  </div>  <p class="calendarevent\_\_info--fixed">    <strong>Time: </strong>  Mon 2019-01-07 12.00  </p>  <p class="calendarevent\_\_info--fixed">  <strong>Lecturer: </strong>  Jiatong Li  </p>  <p class="calendarevent\_\_info--fixed">  <strong>Location: </strong> <span class="value" property="location"> Seminar room Grimeton at CoS, Electrum, elevator B, 4th floor, Isafjordsgatan 22, Kista </span>  </p>  <div class="paragraphs">        <p>Nowadays cloud computing systems handle large amounts of data and process this data across different systems. It is essential to considering data security vulnerabilities and data protection.</p>  <p>One means of decreasing security vulnerabilities is to partition the code into distinct modules and then isolate the execution of the code together with its data. Intel’s Software Guard Extension (SGX) provides security critical code isolation in an enclave. By isolating the code’s execution from an untrusted zone (an unprotected user platform), code integrity and confidentiality are ensured.</p>  <p>Transport Layer Security (TLS) is responsible for providing integrity and confidentiality for communication between two entities. Several TLS libraries support cryptographic functions both for an untrusted zone and an enclave. Different TLS libraries have different performance when used with Intel’s SGX. It is desirable to use the best performance TLS library for specific cryptographic functions.</p>  <p>This thesis describes a performance evaluation several popular TLS libraries performance on Intel SGX. Using the evaluation results and combining several different TLS libraries together, the thesis proposes a new solution to improve the performance of TLS libraries on Intel SGX. The performance is best when invoking the best specific TLS library based upon the data size – as there is a crossover in performance between the two best libraries. This solution also maintains the versatility of the existing cryptographic functions.</p>  <p><strong>Keywords:</strong>&nbsp;<em>Intel SGX, Trusted Execution Environment, TLS libraries</em></p>      </div>            <div class="backLink">  <a href="/en/eecs/aktuellt-pa-eecs/kalender/exjobbspresentatione">To the calendar</a>  </div>  <footer id="articleFooter" class="row">      <div class="publicCategorization col-12">    </div>  <div class="articleFooterDivider"></div>  <div class="col">  <div class="contactInfo">    <span class="label">Page responsible:</span><a href="mailto:communication-support@eecs.kth.se">Webmaster EECS</a>      </div>  <div class="belongsTo">  <span class="label">Belongs to</span>: EECS  </div>  <div class="lastChanged">  <span class="label">Last changed</span>: <span>Jan 02, 2019</span>  </div>  </div>  <div class="col-auto">  <div class="edit">  <a href="/api/liveedit/open?contentId=1.867824" class="editLink" target="\_blank" rel="nofollow" title="Edit this page"></a>  </div>  </div>    </footer>  </article> |

In Swedish:

|  |
| --- |
| <article id="mainContent" role="main" class="article standard calendarevent" vocab="http://schema.org/" typeof="Event" data-cid="1.867824" lang="sv-SE">  <div class="preArticleParagraphs">  <h1 property="name">Optimizing the performance of TLS libraries for SGX</h1>  <div class="lead">        <p>Examensarbete presentation</p>      </div>  </div>  <p class="calendarevent\_\_info--fixed">    <strong>Tid: </strong>  Må 2019-01-07 kl 12.00  </p>  <p class="calendarevent\_\_info--fixed">  <strong>Föreläsare: </strong>  Jiatong Li  </p>  <p class="calendarevent\_\_info--fixed">  <strong>Plats: </strong> <span class="value" property="location"> Seminar room Grimeton at CoS, Electrum, elevator B, 4th floor, Isafjordsgatan 22, Kista </span>  </p>  <div class="paragraphs">        <p>Numera hanterar molnberäkningssystem stora mängder data och bearbetar dessa data över olika system. Det är viktigt att ta itu med datasäkerhetsproblem och dataskydd.</p>  <p>Ett sätt att minska säkerhetsproblem är att partitionera koden i olika moduler och sedan isolera kodens exekvering tillsammans med dess data. Intels Software Guard Extension (SGX) tillhandahåller säkerhetskritisk kodisolering i en enklav. Genom att isolera kodens körning från en otillförlitlig zon (en oskyddad användarplattform) säkerställs kodintegritet och sekretess.</p>  <p>Transport Layer Security (TLS) ansvarar för att ge integritet och konfidentialitet för kommunikation mellan två enheter. Flera TLS-bibliotek stödjer kryptografiska funktioner både för en osäker zon och en enklav. Olika TLS-bibliotek har olika prestanda när de används med Intels SGX. Det är önskvärt att använda TLS-bibliotekets bästa prestanda för specifika kryptografiska funktioner.</p>  <p>Denna avhandling beskriver en prestationsutvärdering av flera populära TLS-bibliotekens prestanda på Intel SGX. Genom att använda utvärderingsresultaten och kombinera flera olika TLS-bibliotek tillsammans, presenterar avhandlingen en ny lösning för att förbättra prestanda för TLS-bibliotek på Intel SGX. Prestandan är bäst när man ringer på det bästa specifika TLS-biblioteket baserat på datastorleken - eftersom det finns en crossover i prestanda mellan de två bästa biblioteken. Denna lösning upprätthåller också mångsidigheten hos de befintliga kryptografiska funktionerna.</p>  <p><strong>Nyckelord:</strong>&nbsp;<em>Intel SGX, Trusted Execution Environment, TLS-bibliotek</em></p>      </div>            <div class="backLink">  <a href="/eecs/aktuellt-pa-eecs/kalender/exjobbspresentatione">Till kalendern</a>  </div>  <footer id="articleFooter" class="row">      <div class="publicCategorization col-12">    </div>  <div class="articleFooterDivider"></div>  <div class="col">  <div class="contactInfo">    <span class="label">Innehållsansvarig:</span><a href="mailto:communication-support@eecs.kth.se">Webmaster EECS</a>      </div>  <div class="belongsTo">  <span class="label">Tillhör</span>: EECS  </div>  <div class="lastChanged">  <span class="label">Senast ändrad</span>: <span>2019-01-02</span>  </div>  </div>  <div class="col-auto">  <div class="edit">  <a href="/api/liveedit/open?contentId=1.867824" class="editLink" target="\_blank" rel="nofollow" title="Redigera denna sida"></a>  </div>  </div>    </footer>  </article> |

# Automating the process after examiner approves a thesis

When a student has submitted the final version of the thesis via the assignment “Examensarbete inlämnande/Final thesis submission” in the Canvas course room.

To automate the assignment of TRITA numbers, a database is used to (1) keep a variable that is atomically incremented when a number in the series is assigned and (2) to keep a record of assignments. The record of assignments is patterned after the spreadsheet that is currently used by the Education office staff.; however, since most of this information is already kept in the gradebook the amount of information to be kept in the database can be minimal.

To access the database we need to install some packages for the OS:

* sudo apt-get install libpq-dev
* sudo apt-get install postgresql-doc-10

and a package for Ruby:

* gem install pg

First we will examine the postgres server that is running in a container.

docker exec -it canvas\_postgres\_1 psql -U postgres

psql (9.5.15)

Type "help" for help.

postgres=# \l

List of databases

Name | Owner | Encoding | Collate | Ctype | Access privileges

---------------------+----------+----------+------------+------------+-----------------------

canvas | postgres | UTF8 | en\_US.utf8 | en\_US.utf8 |

canvas\_test\_rails3\_ | postgres | UTF8 | en\_US.utf8 | en\_US.utf8 |

postgres | postgres | UTF8 | en\_US.utf8 | en\_US.utf8 |

template0 | postgres | UTF8 | en\_US.utf8 | en\_US.utf8 | =c/postgres +

| | | | | postgres=CTc/postgres

template1 | postgres | UTF8 | en\_US.utf8 | en\_US.utf8 | =c/postgres +

| | | | | postgres=CTc/postgres

(5 rows)

From the above we can see that for Canvas there are databases called: canvas and canvas\_test\_rails3\_. In addition, there are some other databases. To this we add a database called trita.

postgres=# CREATE DATABASE trita;

CREATE DATABASE

postgres=# \l

List of databases

Name | Owner | Encoding | Collate | Ctype | Access privileges

---------------------+----------+----------+------------+------------+-----------------------

canvas | postgres | UTF8 | en\_US.utf8 | en\_US.utf8 |

canvas\_test\_rails3\_ | postgres | UTF8 | en\_US.utf8 | en\_US.utf8 |

postgres | postgres | UTF8 | en\_US.utf8 | en\_US.utf8 |

template0 | postgres | UTF8 | en\_US.utf8 | en\_US.utf8 | =c/postgres +

| | | | | postgres=CTc/postgres

template1 | postgres | UTF8 | en\_US.utf8 | en\_US.utf8 | =c/postgres +

| | | | | postgres=CTc/postgres

trita | postgres | UTF8 | en\_US.utf8 | en\_US.utf8 |

(6 rows)

Unfortunately, if we try to connect from a program (Listing 1) running in the VM we will get:

ruby db\_test.rb

Version of libpg: 100006

maguire@chipcanvas:~/utils$ ruby db\_test.rb

Version of libpg: 100006

could not connect to server: No such file or directory

Is the server running locally and accepting

connections on Unix domain socket "/var/run/postgresql/.s.PGSQL.5432"?

Listing : First version of db\_test.rb

|  |
| --- |
| #!/usr/bin/ruby  require 'pg'  puts 'Version of libpg: ' + PG.library\_version.to\_s  begin  con = PG.connect :dbname => 'trita', :user => 'postgres'  puts con.server\_version  rescue PG::Error => e  puts e.message    ensure  con.close if con    end |

This occurs because the postgres server is actually running in a container and not in this VM. We can examiner the container with:

|  |
| --- |
| docker inspect canvas\_postgres\_1  [  {  "Id": "d1384a56dd1bf5d89160b3555796268da1b69999552ff591c98aa2e10cf75869",  "Created": "2018-12-11T22:50:08.042866901Z",  "Path": "docker-entrypoint.sh",  "Args": [  "postgres"  ],  "State": {  "Status": "running",  "Running": true,  "Paused": false,  "Restarting": false,  "OOMKilled": false,  "Dead": false,  "Pid": 4829,  "ExitCode": 0,  "Error": "",  "StartedAt": "2019-02-09T14:16:56.876524023Z",  "FinishedAt": "2019-02-09T14:13:41.319870344Z"  },  "Image": "sha256:4b609bf9668c7cb8d84348b107b618c621a7eccc1d529ffd5b6929c033f3e6c3",  "ResolvConfPath": "/var/snap/docker/common/var-lib-docker/containers/d1384a56dd1bf5d89160b3555796268da1b69999552ff591c98aa2e10cf75869/resolv.conf",  "HostnamePath": "/var/snap/docker/common/var-lib-docker/containers/d1384a56dd1bf5d89160b3555796268da1b69999552ff591c98aa2e10cf75869/hostname",  "HostsPath": "/var/snap/docker/common/var-lib-docker/containers/d1384a56dd1bf5d89160b3555796268da1b69999552ff591c98aa2e10cf75869/hosts",  "LogPath": "/var/snap/docker/common/var-lib-docker/containers/d1384a56dd1bf5d89160b3555796268da1b69999552ff591c98aa2e10cf75869/d1384a56dd1bf5d89160b3555796268da1b69999552ff591c98aa2e10cf75869-json.log",  "Name": "/canvas\_postgres\_1",  "RestartCount": 0,  "Driver": "aufs",  "Platform": "linux",  "MountLabel": "",  "ProcessLabel": "",  "AppArmorProfile": "docker-default",  "ExecIDs": [  "238b39db707f89d0fb755f318cbf052fa2fe7aec682dec8d424fe6f1f5100b61"  ],  "HostConfig": {  "Binds": [  "canvas\_pg\_data:/var/lib/postgresql/data:rw"  ],  "ContainerIDFile": "",  "LogConfig": {  "Type": "json-file",  "Config": {}  },  "NetworkMode": "canvas\_default",  "PortBindings": {},  "RestartPolicy": {  "Name": "",  "MaximumRetryCount": 0  },  "AutoRemove": false,  "VolumeDriver": "",  "VolumesFrom": [],  "CapAdd": null,  "CapDrop": null,  "Dns": [],  "DnsOptions": [],  "DnsSearch": [],  "ExtraHosts": null,  "GroupAdd": null,  "IpcMode": "shareable",  "Cgroup": "",  "Links": null,  "OomScoreAdj": 0,  "PidMode": "",  "Privileged": false,  "PublishAllPorts": false,  "ReadonlyRootfs": false,  "SecurityOpt": null,  "UTSMode": "",  "UsernsMode": "",  "ShmSize": 67108864,  "Runtime": "runc",  "ConsoleSize": [  0,  0  ],  "Isolation": "",  "CpuShares": 0,  "Memory": 0,  "NanoCpus": 0,  "CgroupParent": "",  "BlkioWeight": 0,  "BlkioWeightDevice": null,  "BlkioDeviceReadBps": null,  "BlkioDeviceWriteBps": null,  "BlkioDeviceReadIOps": null,  "BlkioDeviceWriteIOps": null,  "CpuPeriod": 0,  "CpuQuota": 0,  "CpuRealtimePeriod": 0,  "CpuRealtimeRuntime": 0,  "CpusetCpus": "",  "CpusetMems": "",  "Devices": null,  "DeviceCgroupRules": null,  "DiskQuota": 0,  "KernelMemory": 0,  "MemoryReservation": 0,  "MemorySwap": 0,  "MemorySwappiness": null,  "OomKillDisable": false,  "PidsLimit": 0,  "Ulimits": null,  "CpuCount": 0,  "CpuPercent": 0,  "IOMaximumIOps": 0,  "IOMaximumBandwidth": 0,  "MaskedPaths": [  "/proc/acpi",  "/proc/kcore",  "/proc/keys",  "/proc/latency\_stats",  "/proc/timer\_list",  "/proc/timer\_stats",  "/proc/sched\_debug",  "/proc/scsi",  "/sys/firmware"  ],  "ReadonlyPaths": [  "/proc/asound",  "/proc/bus",  "/proc/fs",  "/proc/irq",  "/proc/sys",  "/proc/sysrq-trigger"  ]  },  "GraphDriver": {  "Data": null,  "Name": "aufs"  },  "Mounts": [  {  "Type": "volume",  "Name": "canvas\_pg\_data",  "Source": "/var/snap/docker/common/var-lib-docker/volumes/canvas\_pg\_data/\_data",  "Destination": "/var/lib/postgresql/data",  "Driver": "local",  "Mode": "rw",  "RW": true,  "Propagation": ""  }  ],  "Config": {  "Hostname": "d1384a56dd1b",  "Domainname": "",  "User": "",  "AttachStdin": false,  "AttachStdout": false,  "AttachStderr": false,  "ExposedPorts": {  "5432/tcp": {}  },  "Tty": false,  "OpenStdin": false,  "StdinOnce": false,  "Env": [  "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/lib/postgresql/9.5/bin",  "GOSU\_VERSION=1.10",  "LANG=en\_US.utf8",  "PG\_MAJOR=9.5",  "PG\_VERSION=9.5.15-1.pgdg90+1",  "PGDATA=/var/lib/postgresql/data",  "PG\_COLLKEY\_VERSION=0.5.1"  ],  "Cmd": [  "postgres"  ],  "ArgsEscaped": true,  "Image": "canvas\_postgres",  "Volumes": {  "/var/lib/postgresql/data": {}  },  "WorkingDir": "",  "Entrypoint": [  "docker-entrypoint.sh"  ],  "OnBuild": null,  "Labels": {  "com.docker.compose.config-hash": "960e1938ff64999f8488c7df6c7a952998e16477e28c267755f52e232e3ff23b",  "com.docker.compose.container-number": "1",  "com.docker.compose.oneoff": "False",  "com.docker.compose.project": "canvas",  "com.docker.compose.service": "postgres",  "com.docker.compose.version": "1.22.0"  }  },  "NetworkSettings": {  "Bridge": "",  "SandboxID": "2a14d5bf9b1a23633953e1deb28ce681f78633a428530940c4f1750098bd8649",  "HairpinMode": false,  "LinkLocalIPv6Address": "",  "LinkLocalIPv6PrefixLen": 0,  "Ports": {  "5432/tcp": null  },  "SandboxKey": "/var/snap/docker/321/run/docker/netns/2a14d5bf9b1a",  "SecondaryIPAddresses": null,  "SecondaryIPv6Addresses": null,  "EndpointID": "",  "Gateway": "",  "GlobalIPv6Address": "",  "GlobalIPv6PrefixLen": 0,  "IPAddress": "",  "IPPrefixLen": 0,  "IPv6Gateway": "",  "MacAddress": "",  "Networks": {  "canvas\_default": {  "IPAMConfig": null,  "Links": null,  "Aliases": [  "d1384a56dd1b",  "postgres"  ],  "NetworkID": "3a6597a77b3fc1042422e26cf1a5e8752b9cb5b435b0b08621beeeaf1e021854",  "EndpointID": "2edf27f0db267fa52c158edefd7466cf4c3435c335d0cf528d712a9ac3b2e3e8",  "Gateway": "172.18.0.1",  "IPAddress": "172.18.0.3",  "IPPrefixLen": 16,  "IPv6Gateway": "",  "GlobalIPv6Address": "",  "GlobalIPv6PrefixLen": 0,  "MacAddress": "02:42:ac:12:00:03",  "DriverOpts": null  }  }  }  }  ] |

From the above we can see that the IP address of this postgress server is IPAddress": "172.18.0.3", After chancing the program (Listing 2) one can connect:

ruby db\_test.rb

Version of libpg: 100006

90515ruby db\_test.rb

Version of libpg: 100006

90515

Listing : Second version of db\_test.rb

|  |
| --- |
| #!/usr/bin/ruby  require 'pg'  puts 'Version of libpg: ' + PG.library\_version.to\_s  begin  con = PG.connect :hostaddr => "172.18.0.3", :dbname => 'trita', :user => 'postgres'  puts con.server\_version  rescue PG::Error => e  puts e.message    ensure  con.close if con    end |

Now we can expand this to create a table (deleting it if it already exists) and putting an entry into it as shown in Listing 3. The result is:

ruby db\_test.rb

Version of libpg: 100006

90515

{"id"=>"1"}

1

Listing : Version3 of db\_test.rb

|  |
| --- |
| #!/usr/bin/ruby  require 'pg'  puts 'Version of libpg: ' + PG.library\_version.to\_s  begin  con = PG.connect :hostaddr => "172.18.0.3", :dbname => 'trita', :user => 'postgres'  puts con.server\_version  con.exec "DROP TABLE IF EXISTS eecs\_trita\_for\_thesis\_2018"  rs = con.exec "CREATE TABLE eecs\_trita\_for\_thesis\_2018 (  -- make the 'id' column a primary key; this also creates  -- a UNIQUE constraint and a b+-tree index on the column  id SERIAL PRIMARY KEY,  authors TEXT,  title TEXT,  examiner TEXT)"  rs=con.exec "INSERT INTO eecs\_trita\_for\_thesis\_2018(authors, title, examiner) VALUES ('Ellen FakeStudent', 'A fake title', 'Gerald Q. Maguire Jr.') RETURNING id"  puts(rs[0])  id=rs[0]['id']  puts(id)      rescue PG::Error => e  puts e.message    ensure  con.close if con    end |

Via the psql commands we can see the table has been created and the entry made in Listing 4.

Listing : Contents of the table

|  |
| --- |
| SELECT \* from eecs\_trita\_for\_thesis\_2018;  id | authors | title | examiner  ----+-------------------+--------------+-----------------------  1 | Ellen FakeStudent | A fake title | Gerald Q. Maguire Jr.  (1 row) |

Now the program can be modified to conditionally create the table and add an entry to it as shown in Listing 5.

Listing : Verion 4 of the db\_test.rb program

|  |
| --- |
| #!/usr/bin/ruby  require 'pg'  puts 'Version of libpg: ' + PG.library\_version.to\_s  begin  con = PG.connect :hostaddr => "172.18.0.3", :dbname => 'trita', :user => 'postgres'  puts con.server\_version  #con.exec "DROP TABLE IF EXISTS eecs\_trita\_for\_thesis\_2018"  # create the table if it does not exist  rs = con.exec "CREATE TABLE IF NOT EXISTS eecs\_trita\_for\_thesis\_2018 (  -- make the 'id' column a primary key; this also creates  -- a UNIQUE constraint and a b+-tree index on the column  id SERIAL PRIMARY KEY,  authors TEXT,  title TEXT,  examiner TEXT)"  rs=con.exec "INSERT INTO eecs\_trita\_for\_thesis\_2018(authors, title, examiner) VALUES ('James FakeStudent', 'Another fake title', 'Gerald Q. Maguire Jr.') RETURNING id"  puts(rs[0])  id=rs[0]['id']  puts(id)      rescue PG::Error => e  puts e.message    ensure  con.close if con    end |

Running the program produces the following (and results in the table shown in Listing 6):

ruby db\_test.rb

Version of libpg: 100006

90515

NOTICE: relation "eecs\_trita\_for\_thesis\_2018" already exists, skipping

{"id"=>"2"}

2

Listing : New contents of the table

|  |
| --- |
| SELECT \* from eecs\_trita\_for\_thesis\_2018;  id | authors | title | examiner  ----+-------------------+--------------------+-----------------------  1 | Ellen FakeStudent | A fake title | Gerald Q. Maguire Jr.  2 | James FakeStudent | Another fake title | Gerald Q. Maguire Jr.  (2 rows) |

## Dealing with getting a TRITA number

In this example (using the latest version of announce-presentation.rb), the student James has completed his submission of the final thesis and the examiner as marked it as “Completed” (see in the gradebook in Figure 27).

|  |
| --- |
|  |

Figure : People page

The examiner approves the thesis next and goes to the page of “People” for the course and copies the URL for James (<http://canvas.docker/courses/5/users/12>) and clicks on the “TestButtonLeft” then the examiner will see the page shown in Figure 28.

|  |
| --- |
|  |

Figure : The examiner enters the user's URL

The examiner is now ask about which year the thesis should be reported in (as shown in Figure 29 ).

|  |
| --- |
|  |

Figure : Year thesis is to be reported in

This has been updated to also ask for the language of the thesis, as shown in Figure 30.

|  |
| --- |
|  |

Figure : Thesis year and language

The script processes this input and computes a TRITA string. In this case the school\_prefix is TRITA-EECS-EX and the trita\_string is TRITA-EECS-EX-2019: 1. This is associated with an update of the database as shown in Figure 31.

|  |
| --- |
| trita=# SELECT \* from EECS\_trita\_for\_thesis\_2019;  id | authors | title | examiner  ----+-------------------+--------------------------------+----------------  1 | James FakeStudent | A fake title for a fake thesis | ⚠⚠Dejan Kostic  (1 row) |

Figure : Database after the insertion of the new thesis and assignment of TRITA number)

Note that in the above the (temporary) prefix for the examiner has not been removed. The next step is to generate a cover as will be described in the next subsection.

## Generating a cover for the thesis

There is a thesis cover generator at <https://intra.kth.se/kth-cover>. As this is a form, we will fill in the relevant information and send the form to be processed.

One bit of magic is that we have to include a parameter :model with the value "1337-brynjan!" – otherwise the generator does not work. A test program was written to try all of this, see Listing 7.

Listing : generate\_cover.rb

|  |
| --- |
| #!/usr/bin/ruby  # generate\_cover.rb  #  # A simple test program to create a thesis cover using the KTH cover generator. The resulting PDF file is stored in test.pdf.  #  # G. Q. Maguire Jr.  #  # 2019.02.27  #  require 'json'  require 'httparty'  require 'date'  require 'net/http'  require 'net/http/post/multipart'  uri\_for\_cover = URI("https://intra.kth.se/kth-cover/kth-cover.pdf")  n = Net::HTTP.new(uri\_for\_cover.host, uri\_for\_cover.port)  n.use\_ssl = (uri\_for\_cover.scheme == 'https')  #n.set\_debug\_output($stdout)  parm={:degree=>"second-level-30",  :exam=>4,  :area=>"Information and Communication Technology",  :school => "School of Hard Knocks",  :title=>"A fake title for a fake thesis",  :secondaryTitle=>"A short subtitle",  :author=>["James FakeStudent"],  :trita=>"TRITA-EECS-EX-2019:28",  :model=>"1337-brynjan!"} # this model is important otherwise the generator will not make the page  puts("parm is #{parm}")  req = Net::HTTP::Post::Multipart.new(uri\_for\_cover, parm)  req['Referer']="https://intra.kth.se/kth-cover?l=en"  req['Accept-Encoding']="gzip, deflate, br"  req['Accept-Language']="en-US,en;q=0.9"  req['Accept']="text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8"  res = n.start do |http|  result = http.request(req) # Net::HTTPResponse object  puts("post to create course cover returned #{result}")  puts("result.code is #{result.code}")  puts("Content-Disposition is #{result['Content-Disposition']}")  puts("result.body.length is #{result.body.length}")  file = File.open("test1.pdf", "w")  file.puts("#{result.body}")  file.close  end |

# Adding the cover to the thesis

After making the cover one needs to split the front and back cover apart. This can be done with qpdf as shown in Listing 8

Listing : Splitting the cover apart

|  |
| --- |
| qpdf --split-pages test1.pdf cover\_pages |

If the student’s final thesis is in the file: Fake\_student\_thesis-20190220-for-james.pdf, then we can add the front and back covers as shown in Listing 9.

Listing : Combing file

|  |
| --- |
| qpdf --empty --pages cover\_pages-1 Fake\_student\_thesis-20190220-for-james.pdf cover\_pages-2 -- out.pdf |

With this integrated in announce-presentation.rb we can apply a cover and produce a file, as shown in Figure 32 and Figure 33.

|  |
| --- |
|  |

Figure : Screen output after adding the cover

|  |
| --- |
|  |

Figure : Generated files when making the cover

The next step is to create the MODS file and then upload the MODS file and final thesis with cover to DiVA.

# 10th month reminder

Similar to announcing a oral presentation or approving the final thesis, the 10th month reminder is generated based on the state of a student (i.e., what they have submitted and what has been approved by the examiner or adviser). In this case, once the project plan has been approved the examiner Bertil will see a Canvas Dashboard as shown in Figure 34.

|  |
| --- |
|  |

Figure :Bertil's dashboard after the project plan is approved

Following the method and planning meeting (when the actual starting date is known), the examine can post a calendar reminder for Bertil by going to Bertil’s People page, copying the URL of this page, and then pushing the button is labeled “TestButtonLeft” (as shown in the earlier Figure 11). The examiner fills in the URL as shown in Figure 35.

|  |
| --- |
|  |

Figure : After examiner fills in the URL

After click “Submit”, the examiner will now see a prompt for the date of the reminder as shown in Figure 35.

|  |
| --- |
|  |

If the examiner is using the FireFox brower clicking on the date gives a popup to select a date as shown in Figure 36.

|  |
| --- |
|  |

Figure : Popup calendar for selecting a date

After selecting a date the examiner sees a page similar to Figure 38. Then after submitting this the examiner will see a page as shown in Figure 38.

|  |
| --- |
|  |

Figure : After setting the date

|  |
| --- |
|  |

Figure : After selecting a date

Now Bertil’s Dashboard will look as shown in Figure 39 (note the Month 10 reminder).

|  |
| --- |
|  |

Figure : Bertil's new Dashboard

If Bertil looks at his calendar he will see in January that the 6th has an event, as shown in

|  |
| --- |
|  |

Figure : Zoomed January calendar

If Bertil clicks on the 6th, the calendar will show the view in Figure 41.

|  |
| --- |
|  |

Figure : After clicking on the 6th in the calendar in the corner

Clicking on the event itself reveals the event details, as shown in Figure 42.

|  |
| --- |
|  |

Figure : Event details

Note that to make this posting into a user’ calendar work it was necessary to postfix the request with ?as\_user= xxx (i.e., /api/v1/calendar\_events?as\_user\_id=#{user\_id}) where xxx is the user’s user\_id (in this case “4”) – as this operation is only permitted for the user themselves, hence this postfix makes it appear that the user is making the request (i.e., we *masquerade* as the user).

# Overview of the whole process

Figure 43 shows an overview of the whole process. Note that the assignments (shown as red boxes) can be conditional on a student being in a particular program and specialization; additional assignments (such as the former CSC's group work) can also be added for specific programs/specializations).

|  |
| --- |
|  |

Figure : Overview of whole process

In the figure above solid lines are flows, dashed lines are input, and dotted lines indicate temporal ordering. Items in red are documents produced by the student, green are actions by Education unit, gray are automated actions (with dark gray done earlier, white done now),purple boxes are invoked with TestButtonLef, and violet means to be done.

# Enabling a context sensitive button

As noted previously it would be useful to know what page a user is on when they push the “TestLeftButton”.

Examination of the file app/controllers/application\_controller.rb reveals some interesting keys specicially \_:ping\_url and \_:page\_view\_update\_url that are passed iin the hash to js\_env(hash = {}, overwrite = false).

The file lib/lti/variable\_expander.rb contains the definitions of the variable expansions, for example:

# The title of the context

# @launch\_parameter context\_title

# @example

# ```

# Example Course

# ```

register\_expansion 'Context.title', [],

-> { @context.name },

default\_name: 'context\_title'

After creating a new custom variable:

# Get crumbs array - GQMJR - 2019.03.09

# @launch\_parameter se.kth.chip.crumbs

# @example

# ```

# "This is the current crumbs"

# ```

register\_expansion 'se.kth.chip.crumbs', [],

-> { gqmjr\_crumbs() },

default\_name: 'se\_kth\_chip\_crumbs'

And defining the gqmjr\_crumbs method:

def gqmjr\_crumbs()

str1="crumbs are: "

puts("crumbs are #{str1}")

str1

end

Next the custom variables to pass to the LTI tool are updated with the GUI and the following happens inside the canvas\_web\_1 container:

[d13789bacd36f388e81505ac35f081f4 51f433fb-6b17-4b57-844b-f7885fc7a2ca] [1m[36mSQL SQL (0.6ms)[0m [1m[33mUPDATE "context\_external\_tools" SET "updated\_at" = '2019-03-09 07:09:06.021391', "settings" = '---

:custom\_fields: !ruby/hash:ActiveSupport::HashWithIndifferentAccess

sis\_id: "$Canvas.user.sisSourceId"

user\_sis\_id: "$Person.sourcedId"

context\_source\_id: "$Context.sourcedId"

context\_title: "$Context.title"

context\_id: "$Context.id"

message\_documenttarget: "$Message.documentTarget"

module\_id: "$Canvas.module.id"

module\_id\_item: "$Canvas.moduleItem.id"

chip\_canvas\_module\_id: "$Canvas.module.id"

chip\_canvas\_module\_item\_id: "$Canvas.moduleItem.id"

chip2\_canvas\_module\_id: "${Canvas.module.id}"

chip2\_canvas\_module\_item\_id: "${Canvas.moduleItem.id}"

chip3\_com\_instructure\_context\_label: "$com.instructure.contextLabel"

chip3\_crumbs: "$se.kth.chip.crumbs"

:course\_navigation: !ruby/hash:ActiveSupport::HashWithIndifferentAccess

default: enabled

display\_type: full\_width

enabled: true

text: TestButtonLeft

windowTarget: \_blank

url: http://localhost:3597

visibility: admins

' WHERE "context\_external\_tools"."id" = 3[0m [development:1 master]

[d13789bacd36f388e81505ac35f081f4 51f433fb-6b17-4b57-844b-f7885fc7a2ca] [1m[36mSQL SQL (1.9ms)[0m [1m[33mUPDATE "courses" SET "updated\_at" = '2019-03-09 07:09:06.027818' WHERE "courses"."id" = 5[0m [development:1 master]

[d13789bacd36f388e81505ac35f081f4 51f433fb-6b17-4b57-844b-f7885fc7a2ca] [1m[36mSQL (69.1ms)[0m [1m[35mCOMMIT[0m [development:1 master]

[d13789bacd36f388e81505ac35f081f4 51f433fb-6b17-4b57-844b-f7885fc7a2ca] [1m[36mSQL ContextExternalToolPlacement Load (1.1ms)[0m [1m[34mSELECT "context\_external\_tool\_placements".\* FROM "context\_external\_tool\_placements" WHERE "context\_external\_tool\_placements"."context\_external\_tool\_id" = 3[0m [development:1 master]

[d13789bacd36f388e81505ac35f081f4 51f433fb-6b17-4b57-844b-f7885fc7a2ca] Redis (0.417ms) del accounts/1-20190201131120158519/navigation\_tabs\_key [redis:6379]

[d13789bacd36f388e81505ac35f081f4 51f433fb-6b17-4b57-844b-f7885fc7a2ca] Redis (0.199ms) del rails52:accounts/1-20190201131120158519/navigation\_tabs\_key [redis:6379]

[d13789bacd36f388e81505ac35f081f4 51f433fb-6b17-4b57-844b-f7885fc7a2ca] Completed 200 OK in 183ms (Views: 2.1ms | ActiveRecord: 92.0ms)

When pressing the TestButtonLeft the following is output in the development.log file in the canvas\_web\_1 container:

[d13789bacd36f388e81505ac35f081f4 51f433fb-6b17-4b57-844b-f7885fc7a2ca] Processing by ExternalToolsController#update as JSON

[d13789bacd36f388e81505ac35f081f4 51f433fb-6b17-4b57-844b-f7885fc7a2ca] Parameters: {"external\_tool"=>{"name"=>"announce-presentation", "privacy\_level"=>"anonymous", "consumer\_key"=>"test", "verify\_uniqueness"=>"true", "custom\_fields"=>{"sis\_id"=>"$Canvas.user.sisSourceId", "user\_sis\_id"=>"$Person.sourcedId", "context\_source\_id"=>"$Context.sourcedId", "context\_title"=>"$Context.title", "context\_id"=>"$Context.id", "message\_documenttarget"=>"$Message.documentTarget", "module\_id"=>"$Canvas.module.id", "module\_id\_item"=>"$Canvas.moduleItem.id", "chip\_canvas\_module\_id"=>"$Canvas.module.id", "chip\_canvas\_module\_item\_id"=>"$Canvas.moduleItem.id", "chip2\_canvas\_module\_id"=>"${Canvas.module.id}", "chip2\_canvas\_module\_item\_id"=>"${Canvas.moduleItem.id}", "chip3\_com\_instructure\_context\_label"=>"$com.instructure.contextLabel", "chip3\_crumbs"=>"$se.kth.chip.crumbs"}, "domain"=>"", "url"=>"http://localhost:3597/announce", "description"=>"A test for being able to automate degree project presentation announcements."}, "course\_id"=>"5", "external\_tool\_id"=>"3"}

[d13789bacd36f388e81505ac35f081f4 51f433fb-6b17-4b57-844b-f7885fc7a2ca] [1m[36mSQL SQL (5.4ms)[0m [1m[34mSELECT "pseudonyms"."id" AS t0\_r0, "pseudonyms"."user\_id" AS t0\_r1, "pseudonyms"."account\_id" AS t0\_r2, "pseudonyms"."workflow\_state" AS t0\_r3, "pseudonyms"."unique\_id" AS t0\_r4, "pseudonyms"."crypted\_password" AS t0\_r5, "pseudonyms"."password\_salt" AS t0\_r6, "pseudonyms"."persistence\_token" AS t0\_r7, "pseudonyms"."single\_access\_token" AS t0\_r8, "pseudonyms"."perishable\_token" AS t0\_r9, "pseudonyms"."login\_count" AS t0\_r10, "pseudonyms"."failed\_login\_count" AS t0\_r11, "pseudonyms"."last\_request\_at" AS t0\_r12, "pseudonyms"."last\_login\_at" AS t0\_r13, "pseudonyms"."current\_login\_at" AS t0\_r14, "pseudonyms"."last\_login\_ip" AS t0\_r15, "pseudonyms"."current\_login\_ip" AS t0\_r16, "pseudonyms"."reset\_password\_token" AS t0\_r17, "pseudonyms"."position" AS t0\_r18, "pseudonyms"."created\_at" AS t0\_r19, "pseudonyms"."updated\_at" AS t0\_r20, "pseudonyms"."password\_auto\_generated" AS t0\_r21, "pseudonyms"."deleted\_at" AS t0\_r22, "pseudonyms"."sis\_batch\_id" AS t0\_r23, "pseudonyms"."sis\_user\_id" AS t0\_r24, "pseudonyms"."sis\_ssha" AS t0\_r25, "pseudonyms"."communication\_channel\_id" AS t0\_r26, "pseudonyms"."sis\_communication\_channel\_id" AS t0\_r27, "pseudonyms"."stuck\_sis\_fields" AS t0\_r28, "pseudonyms"."integration\_id" AS t0\_r29, "pseudonyms"."authentication\_provider\_id" AS t0\_r30, "users"."id" AS t1\_r0, "users"."name" AS t1\_r1, "users"."sortable\_name" AS t1\_r2, "users"."workflow\_state" AS t1\_r3, "users"."time\_zone" AS t1\_r4, "users"."uuid" AS t1\_r5, "users"."created\_at" AS t1\_r6, "users"."updated\_at" AS t1\_r7, "users"."avatar\_image\_url" AS t1\_r8, "users"."avatar\_image\_source" AS t1\_r9, "users"."avatar\_image\_updated\_at" AS t1\_r10, "users"."phone" AS t1\_r11, "users"."school\_name" AS t1\_r12, "users"."school\_position" AS t1\_r13, "users"."short\_name" AS t1\_r14, "users"."deleted\_at" AS t1\_r15, "users"."show\_user\_services" AS t1\_r16, "users"."gender" AS t1\_r17, "users"."page\_views\_count" AS t1\_r18, "users"."reminder\_time\_for\_due\_dates" AS t1\_r19, "users"."reminder\_time\_for\_grading" AS t1\_r20, "users"."storage\_quota" AS t1\_r21, "users"."visible\_inbox\_types" AS t1\_r22, "users"."last\_user\_note" AS t1\_r23, "users"."subscribe\_to\_emails" AS t1\_r24, "users"."features\_used" AS t1\_r25, "users"."preferences" AS t1\_r26, "users"."avatar\_state" AS t1\_r27, "users"."locale" AS t1\_r28, "users"."browser\_locale" AS t1\_r29, "users"."unread\_conversations\_count" AS t1\_r30, "users"."stuck\_sis\_fields" AS t1\_r31, "users"."public" AS t1\_r32, "users"."birthdate" AS t1\_r33, "users"."otp\_secret\_key\_enc" AS t1\_r34, "users"."otp\_secret\_key\_salt" AS t1\_r35, "users"."otp\_communication\_channel\_id" AS t1\_r36, "users"."initial\_enrollment\_type" AS t1\_r37, "users"."crocodoc\_id" AS t1\_r38, "users"."last\_logged\_out" AS t1\_r39, "users"."lti\_context\_id" AS t1\_r40, "users"."turnitin\_id" AS t1\_r41 FROM "pseudonyms" LEFT OUTER JOIN "users" ON "users"."id" = "pseudonyms"."user\_id" WHERE "pseudonyms"."id" = 1 LIMIT 1[0m [development:1 master]

[d13789bacd36f388e81505ac35f081f4 51f433fb-6b17-4b57-844b-f7885fc7a2ca] [1m[36mSQL Course Load (0.8ms)[0m [1m[34mSELECT "courses".\* FROM "courses" WHERE (courses.workflow\_state<>'deleted') AND "courses"."id" = 5 ORDER BY "courses"."id" ASC LIMIT 1[0m [development:1 master]

[d13789bacd36f388e81505ac35f081f4 51f433fb-6b17-4b57-844b-f7885fc7a2ca] [1m[36mSQL Enrollment Load (8.8ms)[0m [1m[34mSELECT "enrollments".\* FROM "enrollments" INNER JOIN "enrollment\_states" ON "enrollment\_states"."enrollment\_id" = "enrollments"."id" WHERE "enrollments"."course\_id" = 5 AND (enrollments.workflow\_state<>'deleted') AND "enrollments"."user\_id" = 1 ORDER BY CASE WHEN enrollment\_states.restricted\_access THEN 3 WHEN enrollment\_states.state IN ('active') THEN 0 WHEN enrollment\_states.state IN ('invited', 'creation\_pending', 'pending\_active', 'pending\_invited') THEN 1 WHEN enrollment\_states.state IN ('completed') THEN 2 WHEN enrollment\_states.state IN ('inactive') THEN 3 WHEN enrollment\_states.state IN ('rejected') THEN 4 WHEN enrollment\_states.state IN ('deleted') THEN 5 ELSE 6 END, CASE WHEN enrollments.type IN ('TeacherEnrollment') THEN 0 WHEN enrollments.type IN ('TaEnrollment') THEN 1 WHEN enrollments.type IN ('DesignerEnrollment') THEN 2 WHEN enrollments.type IN ('StudentEnrollment') THEN 3 WHEN enrollments.type IN ('StudentViewEnrollment') THEN 4 WHEN enrollments.type IN ('ObserverEnrollment') THEN 5 ELSE 6 END LIMIT 1[0m [development:1 master]

Note that in the above one case see the request for the custom variable by the LTI tool call:

"chip3\_crumbs"=>"$se.kth.chip.crumbs"

When running the ruby program that implements this LTI tool we see:

ruby announce-presentation.rb

host: canvas.docker

$header: {:Authorization=>"Bearer In0zaZzXd9zFfQ3fa5GN3PwsxzjHoBrLcMpzFznvwrElhj9HAtnw5AmEQruSHGyg", :"Content-Type"=>"application/json", :Accept=>"application/json"}

cycle\_number is 2 and it has class String

school\_acronym is EECS

[2019-03-09 07:36:09] INFO WEBrick 1.4.2

[2019-03-09 07:36:09] INFO ruby 2.5.1 (2018-03-29) [x86\_64-linux-gnu]

== Sinatra (v2.0.5) has taken the stage on 3597 for development with backup from WEBrick

[2019-03-09 07:36:09] INFO WEBrick::HTTPServer#start: pid=10338 port=3597

in route for /

custom\_loc is

request['custom\_loc'] is

request.path\_info is /

request.path is /

request.referer is http://canvas.docker/courses/5/external\_tools/3?display=borderless

params are {"oauth\_consumer\_key"=>"test", "oauth\_signature\_method"=>"HMAC-SHA1", "oauth\_timestamp"=>"1552117620", "oauth\_nonce"=>"JnmCXapG22PZJRq722Pav1yYB4uN1Q0DJdhhJZ7c", "oauth\_version"=>"1.0", "context\_id"=>"35b23d7061f6864f4d5ee67bf552c73079d30577", "context\_label"=>"J5", "context\_title"=>"Test course 5", "custom\_canvas\_enrollment\_state"=>"active", "custom\_chip2\_canvas\_module\_id"=>"$Canvas.module.id", "custom\_chip2\_canvas\_module\_item\_id"=>"$Canvas.moduleItem.id", "custom\_chip3\_com\_instructure\_context\_label"=>"J5", **"custom\_chip3\_crumbs"=>"crumbs are: ",** "custom\_chip\_canvas\_module\_id"=>"$Canvas.module.id", "custom\_chip\_canvas\_module\_item\_id"=>"$Canvas.moduleItem.id", "custom\_context\_id"=>"35b23d7061f6864f4d5ee67bf552c73079d30577", "custom\_context\_source\_id"=>"", "custom\_context\_title"=>"Test course 5", "custom\_message\_documenttarget"=>"iframe", "custom\_module\_id"=>"$Canvas.module.id", "custom\_module\_id\_item"=>"$Canvas.moduleItem.id", "custom\_sis\_id"=>"z0", "custom\_user\_sis\_id"=>"z0", "ext\_roles"=>"urn:lti:instrole:ims/lis/Administrator,urn:lti:instrole:ims/lis/Instructor,urn:lti:role:ims/lis/Instructor,urn:lti:sysrole:ims/lis/SysAdmin,urn:lti:sysrole:ims/lis/User", "launch\_presentation\_document\_target"=>"iframe", "launch\_presentation\_height"=>"400", "launch\_presentation\_locale"=>"en", "launch\_presentation\_return\_url"=>"http://canvas.docker/courses/5/external\_content/success/external\_tool\_redirect", "launch\_presentation\_width"=>"800", "lti\_message\_type"=>"basic-lti-launch-request", "lti\_version"=>"LTI-1p0", "oauth\_callback"=>"about:blank", "resource\_link\_id"=>"35b23d7061f6864f4d5ee67bf552c73079d30577", "resource\_link\_title"=>"announce-presentation", "roles"=>"Instructor,urn:lti:instrole:ims/lis/Administrator", "tool\_consumer\_info\_product\_family\_code"=>"canvas", "tool\_consumer\_info\_version"=>"cloud", "tool\_consumer\_instance\_contact\_email"=>"canvas@canvas.docker", "tool\_consumer\_instance\_guid"=>"Mx0emRDTpd0ZRMuIdpipqIgmGDUsjrosDsiOeJ17:canvas-lms", "tool\_consumer\_instance\_name"=>"chiptest", "user\_id"=>"535fa085f22b4655f48cd5a36a9215f64c062838", "oauth\_signature"=>"DmCQ9DGriWQcHGbECGRPw0MMKWo="}

session['lis\_person\_sourcedid'] is

session['custom\_canvas\_course\_id'] is

session['custom\_sis\_id'] is z0

session['custom\_user\_sis\_id'] is z0

This we can see that the gqmjr\_crumbs method has been called and it returned a string. The next step is to have this return some useful information!

# Jobs

To see what jobs are running in the Canvas instance vist the URL: <http://canvas.docker/jobs>

# Logging into the docker containers

To login as root enter:

docker exec -u 0 -it containername bash

To login as docker into the canvas\_web\_1 docker container you can enter:

If you want to use sftp from the container, you need to install ssh. To do this say:

apt-get install update

apt-get install ssh

In canvas\_web\_1 we can see the environment variables:

docker@f963bfb8c379:/usr/src/app$ env

ENCRYPTION\_KEY=facdd3a131ddd8988b14f6e4e01039c93cfa0160

HOSTNAME=f963bfb8c379

GEM\_HOME=/home/docker/.gem/ruby/2.4.0

TERM=xterm

NGINX\_MAX\_UPLOAD\_SIZE=10g

LC\_ALL=en\_US.UTF-8

LS\_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:\*.tar=01;31:\*.tgz=01;31:\*.arj=01;31:\*.taz=01;31:\*.lzh=01;31:\*.lzma=01;31:\*.tlz=01;31:\*.txz=01;31:\*.zip=01;31:\*.z=01;31:\*.Z=01;31:\*.dz=01;31:\*.gz=01;31:\*.lz=01;31:\*.xz=01;31:\*.bz2=01;31:\*.bz=01;31:\*.tbz=01;31:\*.tbz2=01;31:\*.tz=01;31:\*.deb=01;31:\*.rpm=01;31:\*.jar=01;31:\*.war=01;31:\*.ear=01;31:\*.sar=01;31:\*.rar=01;31:\*.ace=01;31:\*.zoo=01;31:\*.cpio=01;31:\*.7z=01;31:\*.rz=01;31:\*.jpg=01;35:\*.jpeg=01;35:\*.gif=01;35:\*.bmp=01;35:\*.pbm=01;35:\*.pgm=01;35:\*.ppm=01;35:\*.tga=01;35:\*.xbm=01;35:\*.xpm=01;35:\*.tif=01;35:\*.tiff=01;35:\*.png=01;35:\*.svg=01;35:\*.svgz=01;35:\*.mng=01;35:\*.pcx=01;35:\*.mov=01;35:\*.mpg=01;35:\*.mpeg=01;35:\*.m2v=01;35:\*.mkv=01;35:\*.webm=01;35:\*.ogm=01;35:\*.mp4=01;35:\*.m4v=01;35:\*.mp4v=01;35:\*.vob=01;35:\*.qt=01;35:\*.nuv=01;35:\*.wmv=01;35:\*.asf=01;35:\*.rm=01;35:\*.rmvb=01;35:\*.flc=01;35:\*.avi=01;35:\*.fli=01;35:\*.flv=01;35:\*.gl=01;35:\*.dl=01;35:\*.xcf=01;35:\*.xwd=01;35:\*.yuv=01;35:\*.cgm=01;35:\*.emf=01;35:\*.axv=01;35:\*.anx=01;35:\*.ogv=01;35:\*.ogx=01;35:\*.aac=00;36:\*.au=00;36:\*.flac=00;36:\*.mid=00;36:\*.midi=00;36:\*.mka=00;36:\*.mp3=00;36:\*.mpc=00;36:\*.ogg=00;36:\*.ra=00;36:\*.wav=00;36:\*.axa=00;36:\*.oga=00;36:\*.spx=00;36:\*.xspf=00;36:

BUNDLE\_APP\_CONFIG=/home/docker/.bundle

PATH=/home/docker/.gem/ruby/2.4.0/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

PWD=/usr/src/app

DISABLE\_V8\_COMPILE\_CACHE=1

LANG=en\_US.UTF-8

RUBY\_MAJOR=2.4

APP\_HOME=/usr/src/app/

RUBYGEMS\_VERSION=2.6.13

SHLVL=1

HOME=/home/docker

YARN\_VERSION=1.12.3-1

TINI\_VERSION=v0.16.1

RAILS\_ENV=development

DEBIAN\_FRONTEND=noninteractive

LESSOPEN=| /usr/bin/lesspipe %s

VIRTUAL\_HOST=.canvas.docker

LESSCLOSE=/usr/bin/lesspipe %s %s

BUNDLER\_VERSION=1.15.3

\_=/usr/bin/env

OLDPWD=/usr/src/app/log

# Making a SSL Ruby application

In order to avoid the special handling when access an external tool using HTTP rather than HTTPS, I changed announce-presentation.rb to make s-announce-presentation.rb (where the “s-“ stands for secure. In order to do this I made the edit-external-tool-for-course.py program take a new flag “-s” which will change the tool URL into an HTTPS URL if it is an HTTP URL. The tool was run as shown Listing 10.

Listing : Runinng edit tool to secure the external tool's URL

|  |
| --- |
| ./edit-external-tool-for-course.py -s -C 5 3 'ChipIt'  using HTTP for the container environment  Existing tool information for tool\_id=3  {'account\_navigation': None,  'assignment\_edit': None,  'assignment\_menu': None,  'assignment\_selection': None,  'assignment\_view': None,  'collaboration': None,  'consumer\_key': 'test',  'course\_assignments\_menu': None,  'course\_home\_sub\_navigation': None,  'course\_navigation': {'default': 'enabled',  'display\_type': 'full\_width\_in\_context',  'enabled': True,  'label': 'ChipItLeft',  'selection\_height': 400,  'selection\_width': 800,  'text': 'ChipItLeft',  'url': 'http://localhost:3597/announce',  'visibility': 'admins',  'windowTarget': '\_blank'},  'course\_settings\_sub\_navigation': None,  'created\_at': '2019-02-15T15:19:53Z',  'custom\_fields': {'chip2\_canvas\_module\_id': '${Canvas.module.id}',  'chip2\_canvas\_module\_item\_id': '${Canvas.moduleItem.id}',  'chip3\_com\_instructure\_context\_label': '$com.instructure.contextLabel',  'chip\_canvas\_module\_id': '$Canvas.module.id',  'chip\_canvas\_module\_item\_id': '$Canvas.moduleItem.id',  'context\_id': '$Context.id',  'context\_source\_id': '$Context.sourcedId',  'context\_title': '$Context.title',  'message\_documenttarget': '$Message.documentTarget',  'module\_id': '$Canvas.module.id',  'module\_id\_item': '$Canvas.moduleItem.id',  'sis\_id': '$Canvas.user.sisSourceId',  'user\_sis\_id': '$Person.sourcedId'},  'description': 'A test for being able to automate degree project presentation '  'announcements.',  'discussion\_topic\_menu': None,  'domain': None,  'editor\_button': None,  'file\_menu': None,  'global\_navigation': None,  'homework\_submission': None,  'id': 3,  'link\_selection': None,  'migration\_selection': None,  'module\_menu': None,  'name': 'announce-presentation',  'not\_selectable': False,  'post\_grades': None,  'privacy\_level': 'anonymous',  'quiz\_menu': None,  'resource\_selection': None,  'similarity\_detection': None,  'tool\_configuration': None,  'updated\_at': '2019-03-13T12:12:35Z',  'url': 'https://localhost:3597/announce',  'user\_navigation': None,  'vendor\_help\_link': None,  'wiki\_page\_menu': None,  'workflow\_state': 'anonymous'}  {'account\_navigation': None, New state of the tool  'assignment\_edit': None,  'assignment\_menu': None,  'assignment\_selection': None,  'assignment\_view': None,  'collaboration': None,  'consumer\_key': 'test',  'course\_assignments\_menu': None,  'course\_home\_sub\_navigation': None,  'course\_navigation': {'default': 'enabled',  'display\_type': 'full\_width\_in\_context',  'enabled': True,  'label': 'ChipIt',  'selection\_height': 400,  'selection\_width': 800,  'text': 'ChipIt',  'url': 'https://localhost:3597/announce',  'visibility': 'admins',  'windowTarget': '\_blank'},  'course\_settings\_sub\_navigation': None,  'created\_at': '2019-02-15T15:19:53Z',  'custom\_fields': {'chip2\_canvas\_module\_id': '${Canvas.module.id}',  'chip2\_canvas\_module\_item\_id': '${Canvas.moduleItem.id}',  'chip3\_com\_instructure\_context\_label': '$com.instructure.contextLabel',  'chip\_canvas\_module\_id': '$Canvas.module.id',  'chip\_canvas\_module\_item\_id': '$Canvas.moduleItem.id',  'context\_id': '$Context.id',  'context\_source\_id': '$Context.sourcedId',  'context\_title': '$Context.title',  'message\_documenttarget': '$Message.documentTarget',  'module\_id': '$Canvas.module.id',  'module\_id\_item': '$Canvas.moduleItem.id',  'sis\_id': '$Canvas.user.sisSourceId',  'user\_sis\_id': '$Person.sourcedId'},  'description': 'A test for being able to automate degree project presentation '  'announcements.',  'discussion\_topic\_menu': None,  'domain': None,  'editor\_button': None,  'file\_menu': None,  'global\_navigation': None,  'homework\_submission': None,  'id': 3,  'link\_selection': None,  'migration\_selection': None,  'module\_menu': None,  'name': 'announce-presentation',  'not\_selectable': False,  'post\_grades': None,  'privacy\_level': 'anonymous',  'quiz\_menu': None,  'resource\_selection': None,  'similarity\_detection': None,  'tool\_configuration': None,  'updated\_at': '2019-03-13T13:24:09Z',  'url': 'https://localhost:3597/announce',  'user\_navigation': None,  'vendor\_help\_link': None,  'wiki\_page\_menu': None,  'workflow\_state': 'anonymous'} |

Additionally, one need to create certificates for use with SSL. This was done as shown in Listing 11. The pass phrase for the root certificate is canvas.docker and the challenge password is chip.

Listing : Creating the certificates

|  |
| --- |
| maguire@chipcanvas:~$ mkdir certificates\_for\_canvas\_docker  maguire@chipcanvas:~$ cd certificates\_for\_canvas\_docker/  maguire@chipcanvas:~/certificates\_for\_canvas\_docker$ openssl genrsa -des3 -out rootCA.key 2048  Generating RSA private key, 2048 bit long modulus  ............+++  ...............+++  e is 65537 (0x010001)  Enter pass phrase for rootCA.key:  Verifying - Enter pass phrase for rootCA.key:  maguire@chipcanvas:~/certificates\_for\_canvas\_docker$ openssl req -x509 -new -nodes -key rootCA.key -sha256 -days 1024 -out rootCA.pem  Enter pass phrase for rootCA.key:  You are about to be asked to enter information that will be incorporated  into your certificate request.  What you are about to enter is what is called a Distinguished Name or a DN.  There are quite a few fields but you can leave some blank  For some fields there will be a default value,  If you enter '.', the field will be left blank.  -----  Country Name (2 letter code) [AU]:SE  State or Province Name (full name) [Some-State]:Stockholm  Locality Name (eg, city) []:Stockholm  Organization Name (eg, company) [Internet Widgits Pty Ltd]:KTH  Organizational Unit Name (eg, section) []:Chip  Common Name (e.g. server FQDN or YOUR name) []:chip  Email Address []:chip.maguire@gmail.com  maguire@chipcanvas:~/certificates\_for\_canvas\_docker$ ls -als  total 16  4 drwxr-xr-x 2 maguire maguire 4096 Mar 13 11:25 .  4 drwxr-xr-x 27 maguire maguire 4096 Mar 13 11:23 ..  4 -rw------- 1 maguire maguire 1751 Mar 13 11:24 rootCA.key  4 -rw-r--r-- 1 maguire maguire 1415 Mar 13 11:25 rootCA.pem  maguire@chipcanvas:~/certificates\_for\_canvas\_docker$ openssl req -new -sha256 -nodes -out server.csr -newkey rsa:2048 -keyout server.key -config  req: Option -config needs a value  req: Use -help for summary.  maguire@chipcanvas:~/certificates\_for\_canvas\_docker$ openssl req -new -sha256 -nodes -out server.csr -newkey rsa:2048 -keyout server.key  Generating a 2048 bit RSA private key  ....................+++  ...............................+++  writing new private key to 'server.key'  -----  You are about to be asked to enter information that will be incorporated  into your certificate request.  What you are about to enter is what is called a Distinguished Name or a DN.  There are quite a few fields but you can leave some blank  For some fields there will be a default value,  If you enter '.', the field will be left blank.  -----  Country Name (2 letter code) [AU]:SE  State or Province Name (full name) [Some-State]:Stockholm  Locality Name (eg, city) []:Stockholm  Organization Name (eg, company) [Internet Widgits Pty Ltd]:KTH  Organizational Unit Name (eg, section) []:Chip  Common Name (e.g. server FQDN or YOUR name) []:chip  Email Address []:chip.maguire@gmail.com  Please enter the following 'extra' attributes  to be sent with your certificate request  A challenge password []:chip  An optional company name []:  maguire@chipcanvas:~/certificates\_for\_canvas\_docker$ openssl x509 -req -in server.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out server.crt -days 500 -sha256 -extfile v3.ext  x509: Cannot open input file v3.ext, No such file or directory  x509: Use -help for summary.  maguire@chipcanvas:~/certificates\_for\_canvas\_docker$ openssl x509 -req -in server.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out server.crt -days 500 -sha256 -extfile v3.ext  Signature ok  subject=C = SE, ST = Stockholm, L = Stockholm, O = KTH, OU = Chip, CN = chip, emailAddress = chip.maguire@gmail.com  Getting CA Private Key  Enter pass phrase for rootCA.key:  maguire@chipcanvas:~/certificates\_for\_canvas\_docker$ ls -als  total 36  4 drwxr-xr-x 2 maguire maguire 4096 Mar 13 11:35 .  4 drwxr-xr-x 27 maguire maguire 4096 Mar 13 11:23 ..  4 -rw------- 1 maguire maguire 1751 Mar 13 11:24 rootCA.key  4 -rw-r--r-- 1 maguire maguire 1415 Mar 13 11:25 rootCA.pem  4 -rw-r--r-- 1 maguire maguire 17 Mar 13 11:35 rootCA.srl  4 -rw-r--r-- 1 maguire maguire 1432 Mar 13 11:35 server.crt  4 -rw-r--r-- 1 maguire maguire 1078 Mar 13 11:32 server.csr  4 -rw------- 1 maguire maguire 1708 Mar 13 11:31 server.key  4 -rw-r--r-- 1 maguire maguire 222 Mar 13 11:34 v3.ext  maguire@chipcanvas:~/certificates\_for\_canvas\_docker$ sudo cp server.pem /etc/ssl/certs/  [sudo] password for maguire:  cp: cannot stat 'server.pem': No such file or directory  maguire@chipcanvas:~/certificates\_for\_canvas\_docker$ sudo cp server.crt /etc/ssl/certs/  maguire@chipcanvas:~/certificates\_for\_canvas\_docker$ sudo cp server.key /etc/ssl/private/ |

For background about this process see: <http://copypasteprogrammers.com/how-to-get-https-working-on-your-local-development-environment-in-5-minutes-7af615770eec/> and <https://linuxconfig.org/install-apache-on-ubuntu-18-04-bionic-beaver-linux> .

Next we have to tell our browser (in this case Firefox to accept the self-signed certificate (for use with localhost) as shown in Figure 44.

|  |
| --- |
|  |

Figure : Adding exception for localhost

A similar exception was added for canvas.docker resulting in the set of server exceptions shown in Figure 45

|  |
| --- |
|  |

Figure : Exceptions for canvas.docker and localhost servers

Additionally, changes needed to be made to the program to use SSL these were inserted just after the line “set :port, 3597 # an port to use” and are shown in Listing 12, where the sinatra\_ssl.rb code is shown in Listing 13.

Listing : Changes to xxx

|  |
| --- |
| load 'sinatra\_ssl.rb'  set :ssl\_certificate, "/home/maguire/certificates\_for\_canvas\_docker/server.crt"  set :ssl\_key, "/home/maguire/certificates\_for\_canvas\_docker/server.key" |

Listing : sinatra\_ssl.rb based on the answer by Willem Jacob Buys on Jan. 21 2012 at 10:02, see <https://stackoverflow.com/questions/2362148/how-to-enable-ssl-for-a-standalone-sinatra-app>

|  |
| --- |
| require 'webrick/https'  module Sinatra  class Application  def self.run!  certificate\_content = File.open(ssl\_certificate).read  key\_content = File.open(ssl\_key).read  server\_options = {  :Host => bind,  :Port => port,  :SSLEnable => true,  :SSLCertificate => OpenSSL::X509::Certificate.new(certificate\_content),  :SSLPrivateKey => OpenSSL::PKey::RSA.new(key\_content)  }  Rack::Handler::WEBrick.run self, server\_options do |server|  [:INT, :TERM].each { |sig| trap(sig) { server.stop } }  server.threaded = settings.threaded if server.respond\_to? :threaded=  set :running, true  end  end  end  end |

Finally, one can see the result of access the external tool (when clicking the button “ChipIt”, formerly known as “TestButtonLeft”) in terms of network traffic in Figure 46.

|  |
| --- |
|  |

Figure : Network traffic when access the external tool

The output when running the program is shown in Listing 14.

Listing : Output when running the program

|  |
| --- |
| ruby s-announce-presentation.rb  host: canvas.docker  $header: {:Authorization=>"Bearer In0zaZzXd9zFfQ3fa5GN3PwsxzjHoBrLcMpzFznvwrElhj9HAtnw5AmEQruSHGyg", :"Content-Type"=>"application/json", :Accept=>"application/json"}  cycle\_number is 2 and it has class String  school\_acronym is EECS  [2019-03-13 13:27:01] INFO WEBrick 1.4.2  [2019-03-13 13:27:01] INFO ruby 2.5.1 (2018-03-29) [x86\_64-linux-gnu]  [2019-03-13 13:27:01] INFO  Certificate:  Data:  Version: 3 (0x2)  Serial Number:  fb:9d:b1:90:61:e5:c1:f0  Signature Algorithm: sha256WithRSAEncryption  Issuer: C=SE, ST=Stockholm, L=Stockholm, O=KTH, OU=Chip, CN=chip/emailAddress=chip.maguire@gmail.com  Validity  Not Before: Mar 13 11:35:03 2019 GMT  Not After : Jul 25 11:35:03 2020 GMT  Subject: C=SE, ST=Stockholm, L=Stockholm, O=KTH, OU=Chip, CN=chip/emailAddress=chip.maguire@gmail.com  Subject Public Key Info:  Public Key Algorithm: rsaEncryption  Public-Key: (2048 bit)  Modulus:  00:a3:11:e6:ac:9c:81:74:16:65:84:8a:aa:fd:68:  7a:4f:aa:01:19:b5:05:7e:49:58:ba:fa:a8:07:53:  26:e7:9a:0b:c3:1c:fc:5d:3e:6f:47:30:6c:68:bb:  f8:16:63:0e:88:8b:b6:f0:ae:3b:cc:9a:ab:d0:a8:  86:6d:21:71:20:7e:2a:48:81:08:bb:f8:a4:82:2d:  2c:96:ba:43:bf:2c:fd:a1:ea:82:2f:12:35:cc:92:  5e:b7:90:9a:e9:21:29:12:96:90:af:1b:5c:cc:8f:  4a:ac:d7:ec:35:30:09:9c:2b:0e:6f:7d:f0:9e:19:  aa:3c:56:a6:e8:c7:f9:16:51:23:b6:71:34:12:0f:  d1:ab:2c:d3:ba:7a:32:0d:8b:b0:86:b9:d2:ca:fd:  80:16:82:3a:cd:70:df:b4:c9:80:6d:c5:ad:19:d6:  bf:51:90:19:31:e0:33:fd:38:9b:3e:88:0f:e9:f5:  c7:09:17:ba:46:5b:83:d1:4d:95:bd:d1:d9:06:d3:  42:dd:24:b0:44:5b:a8:91:fe:23:9d:54:a5:87:d9:  ff:c1:92:14:a8:9d:a1:4f:3e:94:3c:c0:84:a3:43:  65:6a:ee:a9:91:04:44:fc:64:d2:53:f9:5f:4e:ff:  e9:4b:13:db:bf:95:37:3d:c3:8a:76:35:59:5a:d7:  86:0d  Exponent: 65537 (0x10001)  X509v3 extensions:  X509v3 Authority Key Identifier:  keyid:47:84:70:ED:08:5D:B1:49:15:B2:D6:CE:EE:9E:0B:E3:80:83:40:99  X509v3 Basic Constraints:  CA:FALSE  X509v3 Key Usage:  Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment  X509v3 Subject Alternative Name:  DNS:localhost, DNS:canvas.docker  Signature Algorithm: sha256WithRSAEncryption  1c:56:8b:9b:64:02:5b:32:d6:fd:fe:43:35:86:35:47:70:d1:  9a:40:eb:b6:13:72:2b:e8:bc:39:af:c5:03:79:58:60:eb:6e:  ff:82:b5:03:0c:d9:0d:89:69:06:c8:47:92:4d:48:8f:61:be:  33:2c:12:7f:32:96:ba:c8:a4:96:20:76:42:cf:98:30:64:a3:  f2:a7:62:14:8b:23:9e:c5:ad:b1:04:13:d4:4d:55:ab:5c:88:  3c:5b:e7:7c:85:78:de:4f:5a:00:e3:b6:d9:73:a4:4d:41:50:  68:cd:5e:7a:7f:66:43:86:2b:0c:89:7f:fc:3c:3b:1c:24:bf:  c2:aa:3d:c2:ba:2b:61:b4:03:66:ef:bc:2a:5e:bc:42:f5:f2:  d3:5b:6a:2f:54:52:d0:4a:1e:06:9b:d3:55:4e:3f:1d:e1:26:  38:c8:70:48:fa:5b:cb:1a:83:3d:e2:5d:96:c5:b3:88:01:04:  cb:5e:4e:41:9f:19:f4:6b:b9:73:b2:e6:1e:1e:a0:fe:cd:9f:  5e:df:70:5d:9c:28:90:4e:53:85:93:18:61:33:12:81:4c:af:  79:cc:a8:3d:a7:43:06:96:08:c7:6f:bb:ca:39:00:5c:13:0d:  9d:92:75:43:1a:10:7a:97:6d:06:be:30:00:6d:fe:c5:ed:b6:  53:b8:09:f2  [2019-03-13 13:27:01] INFO WEBrick::HTTPServer#start: pid=19869 port=3597  in route for /  custom\_loc is  request['custom\_loc'] is  request.path\_info is /announce  request.path is /announce  request.referer is https://canvas.docker/courses/5/external\_tools/3?display=borderless  params are {"oauth\_consumer\_key"=>"test", "oauth\_signature\_method"=>"HMAC-SHA1", "oauth\_timestamp"=>"1552483767", "oauth\_nonce"=>"ppX7zVKIpj3K9Dqtq9nOFcin0gqzwFM0p3E2R1XdQCM", "oauth\_version"=>"1.0", "context\_id"=>"35b23d7061f6864f4d5ee67bf552c73079d30577", "context\_label"=>"J5", "context\_title"=>"Test course 5", "custom\_canvas\_enrollment\_state"=>"active", "custom\_chip2\_canvas\_module\_id"=>"$Canvas.module.id", "custom\_chip2\_canvas\_module\_item\_id"=>"$Canvas.moduleItem.id", "custom\_chip3\_com\_instructure\_context\_label"=>"J5", "custom\_chip\_canvas\_module\_id"=>"$Canvas.module.id", "custom\_chip\_canvas\_module\_item\_id"=>"$Canvas.moduleItem.id", "custom\_context\_id"=>"35b23d7061f6864f4d5ee67bf552c73079d30577", "custom\_context\_source\_id"=>"", "custom\_context\_title"=>"Test course 5", "custom\_message\_documenttarget"=>"iframe", "custom\_module\_id"=>"$Canvas.module.id", "custom\_module\_id\_item"=>"$Canvas.moduleItem.id", "custom\_sis\_id"=>"z0", "custom\_user\_sis\_id"=>"z0", "ext\_roles"=>"urn:lti:instrole:ims/lis/Administrator,urn:lti:instrole:ims/lis/Instructor,urn:lti:role:ims/lis/Instructor,urn:lti:sysrole:ims/lis/SysAdmin,urn:lti:sysrole:ims/lis/User", "launch\_presentation\_document\_target"=>"iframe", "launch\_presentation\_height"=>"400", "launch\_presentation\_locale"=>"en", "launch\_presentation\_return\_url"=>"https://canvas.docker/courses/5/external\_content/success/external\_tool\_redirect", "launch\_presentation\_width"=>"800", "lti\_message\_type"=>"basic-lti-launch-request", "lti\_version"=>"LTI-1p0", "oauth\_callback"=>"about:blank", "resource\_link\_id"=>"35b23d7061f6864f4d5ee67bf552c73079d30577", "resource\_link\_title"=>"announce-presentation", "roles"=>"Instructor,urn:lti:instrole:ims/lis/Administrator", "tool\_consumer\_info\_product\_family\_code"=>"canvas", "tool\_consumer\_info\_version"=>"cloud", "tool\_consumer\_instance\_contact\_email"=>"canvas@canvas.docker", "tool\_consumer\_instance\_guid"=>"Mx0emRDTpd0ZRMuIdpipqIgmGDUsjrosDsiOeJ17:canvas-lms", "tool\_consumer\_instance\_name"=>"chiptest", "user\_id"=>"535fa085f22b4655f48cd5a36a9215f64c062838", "oauth\_signature"=>"LFrx3qyFKk1nGn7iAozZBQKliZc="}  session['lis\_person\_sourcedid'] is  session['custom\_canvas\_course\_id'] is  session['custom\_sis\_id'] is z0  session['custom\_user\_sis\_id'] is z0  ::1 - - [13/Mar/2019:13:29:30 +0000] "POST /announce HTTP/1.1" 303 - 0.0055  ::1 - - [13/Mar/2019:13:29:30 UTC] "POST /announce HTTP/1.1" 303 0  https://canvas.docker/courses/5/external\_tools/3?display=borderless -> /announce  In /getURL  ::1 - - [13/Mar/2019:13:29:30 +0000] "GET /getURL HTTP/1.1" 200 590 0.0003  ::1 - - [13/Mar/2019:13:29:30 UTC] "GET /getURL HTTP/1.1" 200 590  https://canvas.docker/courses/5/external\_tools/3?display=borderless -> /getURL |

By changing the POST ‘/annouce’ route as shown in Listing 15 and changing attributes of the tool to set course\_navigation[windowTarget] to ‘\_sef’ rather than ‘\_blank’ the question is ask in an iframe, as shown in Figure 47.

Listing : New POST '/annouce' router

|  |
| --- |
| post '/announce' do  puts("in POST route for /announce")  custom\_loc=params['custom\_loc']  puts("custom\_loc is #{custom\_loc}")  @str1=request['custom\_loc']  puts("request['custom\_loc'] is #{@str1}")  path\_info=request.path\_info  puts("request.path\_info is #{path\_info}")  path=request.path  puts("request.path is #{path}")  ref=request.referer  puts("request.referer is #{ref}")  puts "params are #{params}"  begin  signature = OAuth::Signature.build(request, :consumer\_secret => $oauth\_secret)  signature.verify() or raise OAuth::Unauthorized  rescue OAuth::Signature::UnknownSignatureMethod,  OAuth::Unauthorized  return %{unauthorized attempt. make sure you used the consumer secret "#{$oauth\_secret}"}  end  # store the relevant parameters from the launch into the user's session, for  # access during subsequent http requests.  # note that the name and email might be blank, if the tool wasn't configured  # in Canvas to provide that private information.  %w(lis\_outcome\_service\_url lis\_result\_sourcedid lis\_person\_name\_full lis\_person\_contact\_email\_primary  lis\_person\_sourcedid custom\_canvas\_course\_id custom\_canvas\_user\_id custom\_sis\_id  custom\_user\_sis\_id  ).each { |v| session[v] = params[v] }  puts "session['lis\_person\_sourcedid'] is #{session['lis\_person\_sourcedid']}"  puts "session['custom\_canvas\_course\_id'] is #{session['custom\_canvas\_course\_id']}"  puts "session['custom\_sis\_id'] is #{session['custom\_sis\_id']}"  puts "session['custom\_user\_sis\_id'] is #{session['custom\_user\_sis\_id']}"  #gqmjr  <<-HTML  <form action="/gotURL" method="post">  <h2>Enter URL of student's People page</span> | <span lang="sv">Ange webbadressen till studentens sidan från Personer i kursen?</span></h2>  <input name='s\_URL' type='text' width='1000' id='s\_URL' />  <br><button type="cancel" onclick="window.location='getURL';return false;">Cancel</button>  <input type='submit' value='Submit' />  </form>  HTML  # redirect to("/getURL")  end |

|  |
| --- |
|  |

Figure : Using a target of \_self rather than \_blank

We can now see that there are breadcrumbs available, as shown in Listing 15.

|  |
| --- |
|  |

Figure : The breadscrumbs are visible

1. <https://community.canvaslms.com/docs/DOC-10891-4214418311> states “If custom cascading style sheets (CSS) or JavaScript (JS) override files have been enabled for your account, the Theme Editor sidebar displays an Edit tab [1] and an Upload tab [2]. The Theme Editor defaults to the Edit tab and shows all the Theme Editor components. The Upload tab allows you to upload custom files.” [↑](#footnote-ref-1)