



[A] Production server with Real-Time Web

In this assignment, you will be writing a web application where you have to include real-time web technologies. You should publish your application on a real (public) production server.

The idea behind the application is that you should be able to list issues from a GitLab repository through the GitLab REST API and GitLab's Webhooks.

Goals for this assignment

The assignment aims to give the student practical and theoretical experience about developing real-time web applications through Web socket and webhooks. The student should also get practical experience on how to put the built web application into production in a secure way.

Resources

Use these resources to solve the assignment.

- [GitLab REST API](#)
- [GitLab Webhooks](#)

Grading

This assignment is graded as F, Fx, E-A and it is worth 4 credit/hp.

Get going

Get going with the assignment by cloning and preparing the assignment repo.



1. You have a git repo on GitLab named "b3-production", available below your Inu student acronym in the course at GitLab. Start of by cloning the repo to your computer ("xxx" is your Inu-username):

```
# Using ssh
git clone git@gitlab.lnu.se:1dv528/student/xxx/b3-production.git

# Move into the repo
cd b3-production
```

Maintainer group

1. You have a group on GitLab called "maintainer" where you have the role "maintainer" which is needed to create GitLab WebHooks. Use that group and create a repo to which you may connect WebHooks. Make your application listening to those webhooks.

Requirements



The image above tries to explain the application flow in this assignment.

1. When a client connects to the application it will contact GitLab through their REST API and fetch all created issues on your repository. The response will be in JSON, which is good because we work with Node.js.
2. When your application gets the issue list from GitLab, you should use that to render the HTML-page for the client along with the JavaScript needed for the client.
3. One thing the client script needs to do is, for example, the ability to connect to your server's WebSocket channel.
4. When a new "issue-event" happens on GitLab they will fire a (by you) registered HTTP POST which should point to your application.



Other requirements

- Design your application with security in mind.
 - For instance, your code must check that the webhook POST comes from GitLab.
- Along with your code, and installation scripts you should also commit an assignment report that answers some questions (see below).
- You are free to choose and use third-party packages to solve the problem but they should be mentioned and motivated in your assignment report.

To handle API keys

To be able to request data from GitLab's web API you must get an API token that identifies yourself. Send the key in the Authorization header in every request so that GitLab could authorize the request. Please notice that this key is personal to your GitLab account. That means that the key should never be stored in a repository or shared with anyone (not even the teachers).

When you register a callback for the webhook, you should also define a secret so that you can validate the hook-requests to your application. This is also one thing to keep secret.

You can solve this in a couple of ways. In this assignment, you should **use environment variables**. This way you can read from the environment variables in your application and add it when you start your node-application, for example using dotenv.

Production of application

The application shall be running on a public web server in a production environment. Each student will get their virtual public server (see the production part of this course) but feel free to choose your provider. When doing the examination the application should be able to run on your production server (we will not run it locally) therefore you must add the URL to the application in your repository README.

Production requirements



- The server should only listen to port 80, 443 and 22(ssh). That means you should not be able to make requests directly to your node application and that all requests go through the reversed proxy.
- The Node.js application should be running through PM2 and be in production mode.
- If you use some kind of installation script on your production server, this should be added to your repository.

Extra features [optional]

For those of you aiming for higher grades in this course, there will be some suggestions for extra features that you could implement. You also need to document each feature and describe what you did and how it works.

1. First of all - You probably have other ideas, feel free to try them.
2. Make a good looking and user friendly user interface.
3. Implement a richer web client that can control the issues through the application (closing issues, adding comments, etc.). Think about security!
4. Add more integrations than issues, for example commits, tags and releases.
5. Add some kind of custom authentication before the user can enter the application.
6. Do an authentication through [GitLab OAuth provider](#) instead of the Basic Authentication. This way a user could log into your applications through their OAuth credentials and will see all of their resources. The user could then choose what repository to watch and the server application creates the webhook through the web API.

Grading

These are a few notes on how your submission is graded.

Grade E, D

Fulfill all mandatory features and requirements to a satisfactory level where they work as expected.



Grade D, C

As above and your application, documentation and presentation has a high level of quality.

The outcome is visual appealing and has a high level of usability.

You have implemented and documented one or a few optional requirements.

Grade B, A

Your application, documentation and presentation has an excellent level of quality, usability and fulfillment of all the features and their requirements.

You have implemented and explained some optional/added feature.

You have implemented and documented several optional requirements and perhaps even the GitLab OAuth.

Presentation

You are to record a video presentation of your project where you show how your project implements the features and fulfills the requirements. You should also present a few highlights from your source code and elaborate on why you feel the code is good/bad.

The video scene should contain a cam where your face is visible.

The tool OBS is a free and opensource tool you can use to record the presentation video.

The video should be 5-7 minutes.

Submission

This is how to submit this assignment.

1. Add a tag `v1.0.0` to your repo when you are done. If you make updates then add another tag like `v1.0.1` or `v1.1.0` and so on.
2. Ensure you have committed and pushed all your changes, including the tags, to GitLab.



4. Create an issue on the repo and answer the following questions in it. Write freely with 15 to 30 sentences of text.
1. Add the URL to your presentation video, you may store it non-public on youtube, google drive, dropbox or equal.
 2. Describe what you have done to make your application secure, both in code and when configuring your application server.
 3. Describe the following parts, how you are using them, and what their purpose is in your solution:
 1. Reversed proxy
 2. Process manager
 3. TLS certificates
 4. Environment variables
 4. What differs in your application when running it in development from running it in production?
 5. Which extra modules did you use in the assignment? Motivate the use of them, and how you have to make sure that they are secure enough for production.
 6. Have you implemented any extra optional features that could motivate a higher grade of this assignment? If so, describe them - one by one and be verbose on how you solved them.
 7. Are you satisfied with your application? Does it have some improvement areas? What are you especially satisfied with?
 8. What is your TIL for this course part?
 9. What is your overall TIL for this course?
5. When you are done, assign the issue to the teacher, to show that you are ready for grading. The teacher will be notified by mail.



