
correctexam

Release latest

Barais Olivier

Nov 05, 2022

HOW TO:

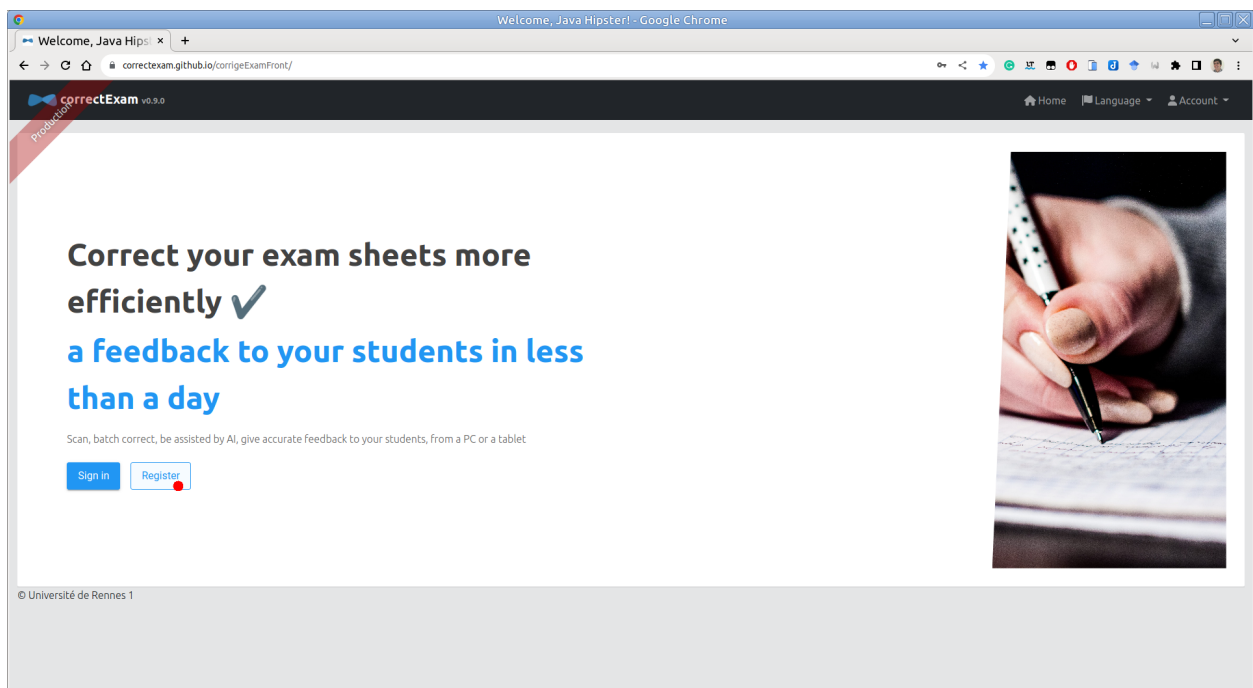
1	How to ?	1
1.1	For users	1
1.2	For developers	7
2	AMD64	9
2.1	Build	9
2.2	Deploy everything on your own infrastructure	10
2.3	Or Deploy the database and the backend on your own infrastructure and the frontend on a CDN . . .	13
3	Build and deploy on raspberry PI (arm64)	15
3.1	Install support of cross compile on your machine	15
3.2	Build	15
3.3	Deploy on your raspberry 4	16
4	Create a release on docker hub	21

HOW TO ?

1.1 For users

1.1.1 How to create an account on the platform?

You can go to the [platform](#), click on **register**, fill in some information. You will receive an email to validate your account.



The screenshot shows a web browser window with the URL `correctexam.github.io/corrigExamFront/`. The page features a dark header with the **correctExam** logo and navigation links for Home, Language, and Account. The main content area has a white background with the text: "Correct your exam sheets more efficiently ✓ a feedback to your students in less than a day". Below this text is a small line of descriptive text: "Scan, batch correct, be assisted by AI, give accurate feedback to your students, from a PC or a tablet". There are two buttons: "Sign in" and "Register". On the right side, there is a photograph of a hand holding a pen over an exam sheet. A red diagonal banner in the top left corner of the page reads "Production". At the bottom left, there is a copyright notice: "© Université de Rennes 1".

Registration

Username
barais

Email
Your email

New password

Password strength

New password confirmation
Confirm the new password

Register

If you want to [sign in](#).

© Université de Rennes 1

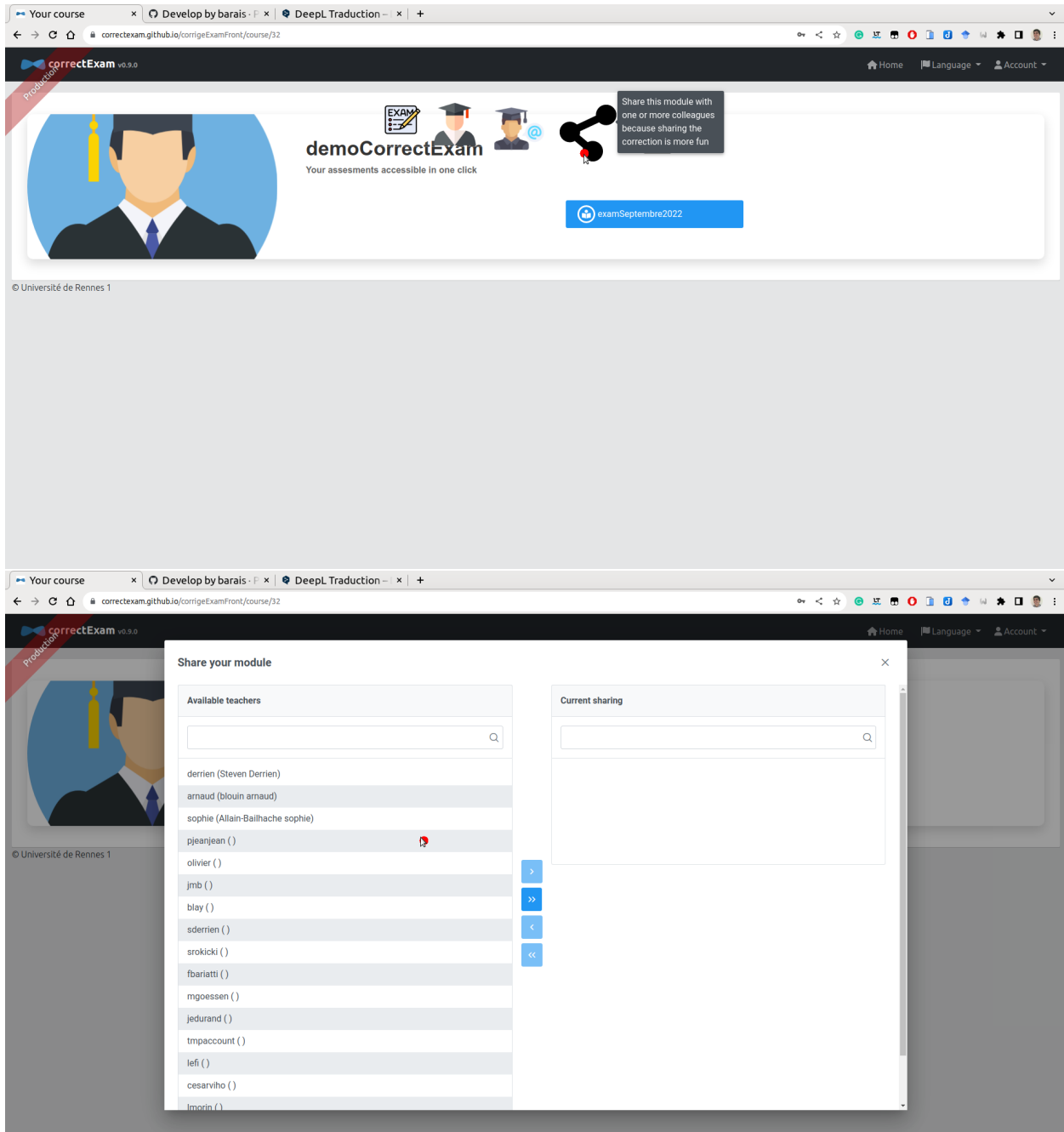
You can then fill in your account information in the menu. *Account -> Settings*

1.1.2 How to create a module?

Once authenticated, on the main page, click on the + symbol *create a course*. It is required to give a name to a module.

1.1.3 Can I share a correction with one or more colleagues?

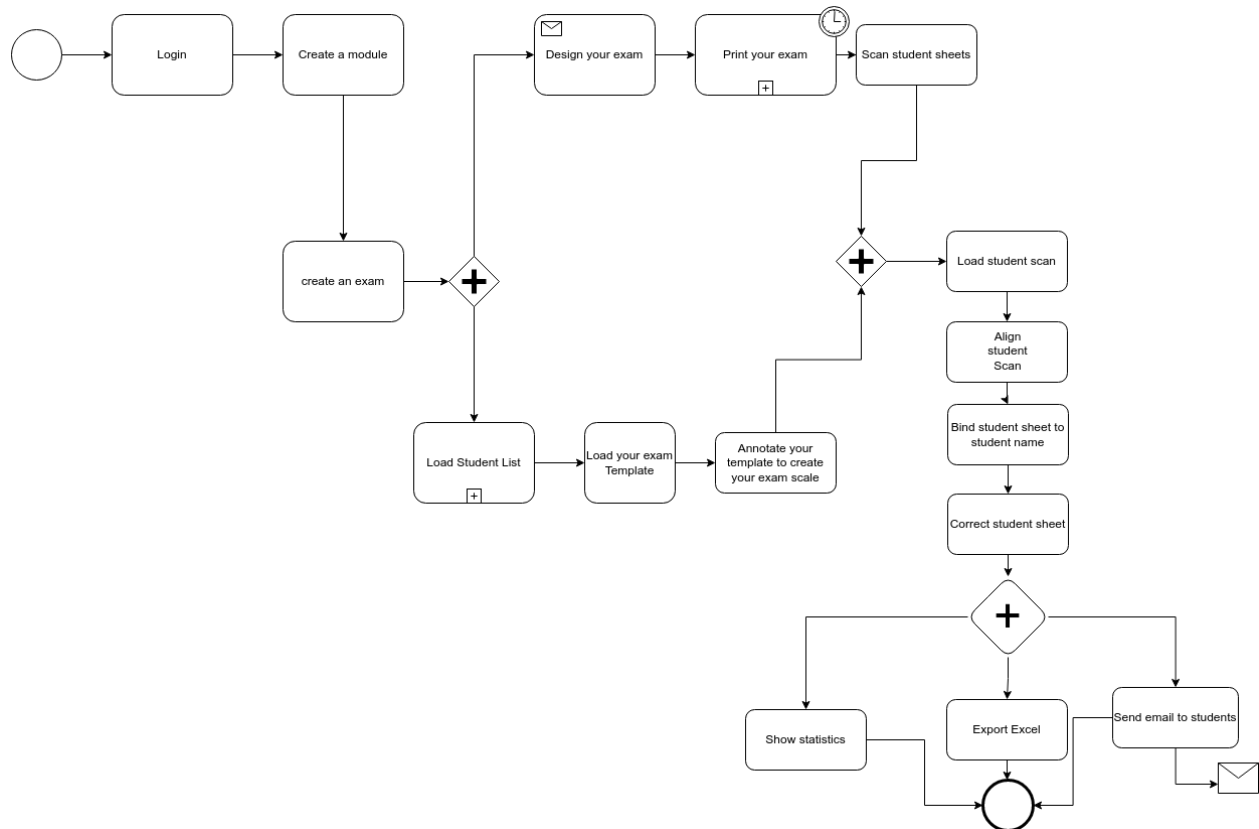
When you click on the module, you will have access in the list of actions to a shared action that allows you to share this module with one or more colleagues. These colleagues will then see this module in the list of their modules.



1.1.4 Can I limit the rights of a colleague so that he can only access a subset of the questions?

No, a colleague, we can trust him ;). We do not implement any specific RBAC rules per exam.

1.1.5 Do you have a diagram of the main steps to do to use this application?



1.1.6 How to create an exam?

To create an exam, once you enter the page of a module, it is possible to create a new exam with the command carousel.

1.1.7 Where can I find some templates for Word latex and libreoffice?

In the view that allows you to create an exam, you have access to a number of templates to create your exam. The philosophy of the application is to let each teacher create his exam with the tool he likes.

1.1.8 Why am I forced to upload a list of students?

To correct, we associate each answer key with a student. This list is required to do the assignment task.

1.1.9 Do I have to delete a student who did not compose?

No, it will be marked as ABI by default.

1.1.10 Can you explain the different types of questions that are available?

For the moment, there are roughly four types of questions.

- **MCQs** for which the application provides marking assistance.
- The **DIRECT** scoring (*Manual and Direct*) for which the teacher manually scores the answers to this question.
- **POSITIVE** scoring (*Manual and POSITIVE*). This remains an element for which the teacher can define a set of comments along the way that gives points to the answers for this question (we start from zero). The total number of points obtained cannot exceed the maximum number of points associated with this question.
- The **NEGATIVE** notation (*Manual and NEGATIVE*). This is an element for which the corrector can define a set of comments along the way that removes points from the answer in question (we start from the maximum number of points possible for this question). The total number of points obtained cannot go below zero.

These types of questions will be enriched in the future. Please provide your great idea.

1.1.11 Can I change the type of question when the correction of a question has started?

Honestly, not recommended because not tested.

1.1.12 How to clean the students' scan if pages are missing, duplicate page, flipped page...?

Use [pdfarranger](#). It is great tool for this task.

1.1.13 Can I reload a clean scan (see previous question) and realign it if I already start to correct an exam?

Yes, no problem, if you share the correction with colleagues or with different devices, you can force the upload and download to the server later.

1.1.14 What happens if my template has an odd number of pages (e.g. 3 pages) and the scan of the student sheets is a multiple of 2 (e.g. 4 page per student)

For the moment, it is not managed correctly at the moment of the alignment. It will be necessary to remove the blank pages from the student scans beforehand with pdf manipulation tools like `pdftk` or `pdfarranger`.

1.1.15 Can I test an instance of this application on my own infrastructure/laptop (I want to keep my data private) ?

Yes please go to the documentation for developers. We provide scripts to deploy this application on any type of infrastructure from a powerful server with K8S to a raspberry 4.

You can also easily test it locally. You just need docker.

```
git clone -b develop https://github.com/correctexam/corrigeExamBack
cd corrigeExamBack/src/main/docker
docker-compose -f app.yml build --no-cache back front
docker-compose -f app.yml up
```

the application is then available on <http://localhost:9000> the phpmyadmin part is available on <http://localhost:91> the fake mail server part is available on <http://localhost:9000/maildev/>

For information:

1. If you want to connect to a real mail server type share. you can have an overview of the quarkus properties as a comment in the *app.yaml* file.
2. If you want to change the ports
 - for phpadmin,
 - for the host port you have to change it in the *app.yml* file
 - for the internal port, you have to change it both in the *app.yml* file and the *myadmin.conf* file (nginx container file)
 - for the application
 - for the host port, you have to change it in the *app.yml*.
 - for the internal port, you have to change it in the *app.yml* and in the *exam.conf* file (file of the nginx container) for the front and update the quarkus properties if you want to change the internal port of the back (no real reason)
1. If you want to set up reverse proxy according to a domain name, everything will happen in the files *exam.conf* and *myadmin.conf* but there is also to update the external url in the application, it's a quarkus property in the *app.yml* file.

1.1.16 How do I scan students' sheets?

We recommend using grayscale at 150 DPI to keep the file size reasonable but no problem with a color scan at 300 DPI if necessary. Avoid pure black and white which could be detrimental to the algo that recognizes names/first names/ids (INE) of students

1.1.17 Can I compose a statement with another tool than word, excel or latex?

Of course, the only important point is a circle marker in the corners to facilitate alignment and light gray squares to put the names, first names and ids (INE) of students.

1.1.18 What to do when I encounter a bug in the application?

Use the github ticket system on the project <https://github.com/correctexam/corrigeExamFront>

1.2 For developers

1.2.1 how to contribute ?

We try to follow a gitflow workflow to integrate new contributions. Do not hesitate to read the [gitflow documentation](#) and do pull request. We are welcome to external contributions so do not hesitate even if you are not sure of your new contributions .

Read the following guide to setup your own development environment.

1.2.2 how to setup your own development environment?

Read the following guide to setup your own development environment.

2.1 Build

2.1.1 Backend

If you want to build it manually, you can just run the following command:

```
git clone https://github.com/correctexam/corrigeExamBack
cd corrigeExamBack
mvn -B package --file pom.xml -Pnative
docker build -f src/main/docker/Dockerfile.native -t barais/correctexam-back:manifest-
↳amd64 --build-arg ARCH=amd64/ .
```

The backend is also built automatically using github action. You can have access to the backend image within `docker hub`

OR

```
#if you install the quarkus cli
git clone https://github.com/correctexam/corrigeExamBack
cd corrigeExamBack
quarkus build --native --no-tests -Dquarkus.native.container-build=true
docker build -f src/main/docker/Dockerfile.native -t barais/correctexam-back:manifest-
↳amd64 --build-arg ARCH=amd64/ .
```

2.1.2 Front

Without docker Just clone the project

update `webpack/environment.js` with your domain name.

```
# require nodejs v16 you can install it using nvm (https://github.com/nvm-sh/nvm)
git clone https://github.com/correctexam/corrigeExamFront
cd corrigeExamFront
# update webpack/environment.js with your domain names
npm install
npm run webapp:build:prod
## You can be inspired by webapp:build:prodgithubpage task if you want to manage a prefix.
↳for your webapp.
## if you want to deploy on github page, you can be inspired by the provided github.
↳action
```

Using docker

To build the front, we provide a simple docker file.

update webpack/environment.js with your domain name.

```
git clone https://github.com/correctexam/corrigeExamFront
cd corrigeExamFront
# update webpack/environment.js with your domain name
sudo docker build -f src/main/docker/Dockerfile -t barais/correctexam-front:manifest-
↪amd64 --build-arg ARCH=amd64/ .
# OR
sudo docker buildx build -f src/main/docker/Dockerfile --push --platform linux/arm64,
↪linux/amd64 --tag barais/correctexam-front .
```

You will obtain a nginx with only the js, html and js. You have to mount the configuration if you want to manage proxy to the backend routes. I would prefer to use bunkerized nginx for the routing.

2.2 Deploy everything on your own infrastructure

```
version: '2'
services:
  correctexam-back:
    image: barais/correctexam-back:manifest-amd64
    volumes:
# Path for
  - /tmp/files:/tmp/files:rw
    restart: always
    ports:
  - 8080:8080
# All quarkus configuration parameters (knobs) could be override through command line.
↪You could also use different options to update the configuration parameters. https://
↪quarkus.io/guides/config-reference
    command: ./application -Dquarkus.http.host=0.0.0.0 -Dquarkus.datasource.
↪username=root -Dquarkus.datasource.password='' -Dquarkus.datasource.jdbc.
↪url=jdbc:mysql://correctexam-mysql:3306/correctexam?useUnicode=true&
↪characterEncoding=utf8&useSSL=false -Dquarkus.http.cors=true -Dquarkus.http.cors.
↪origins=https://correctexam.github.io -Dquarkus.http.cors.methods=GET,PUT,POST,DELETE,
↪PATCH,OPTIONS -Dquarkus.http.cors.headers=accept,origin,authorization,content-type,x-
↪requested-with -Dquarkus.http.cors.exposed-headers=Content-Disposition -Dquarkus.http.
↪cors.access-control-max-age=24H -Dquarkus.mailer.from=olivier.barais@univ-rennes1.fr -
↪Dquarkus.mailer.host=partage.univ-rennes1.fr -Dquarkus.mailer.port=587 -Dquarkus.
↪mailer.ssl=false -Dquarkus.mailer.username=olivier.barais@univ-rennes1.fr -Dquarkus.
↪mailer.password=TOCHANGE -Djhipster.mail.base-url=https://correctexam.github.io/
↪corrigeExamFront
  correctexam-mysql:
    image: mysql:8.0.20
    volumes:
  - ../../resources/db/migration/:/docker-entrypoint-initdb.d
    environment:
  - MYSQL_USER=root
  - MYSQL_ALLOW_EMPTY_PASSWORD=yes
```

(continues on next page)

(continued from previous page)

```

- MYSQL_DATABASE=correctexam
  command: mysqld --lower_case_table_names=1 --skip-ssl --character_set_server=utf8mb4
↳ --explicit_defaults_for_timestamp
#   ports:
#     - 3308:3306
front:
  image: barais/correctexam-front:manifest-amd64
#   ports:
#     - 90:80
  volumes:
    - ./exampleconf/exam.conf:/etc/nginx/conf.d/exam.conf
    - ./exampleconf/nginx.conf:/etc/nginx/nginx.conf:ro

```

exam.conf and **nginx.conf** could be something like that (you have to update the server name)

exam.conf

```

server {
  listen      80;
  listen     [::]:80;
  # server name to change based on your own domain name for your front
  server_name correctexam.barais.fr;

  #charset koi8-r;
  #access_log /var/log/nginx/host.access.log main;

  location /api {
    proxy_pass http://correctexam-back:8080/api;
    proxy_set_header Host $http_host;
  }

  location /api {
    include proxy_params;
    proxy_pass http://correctexam-back:8080/api;
  }

  location /management {
    include proxy_params;
    proxy_pass http://correctexam-back:8080/management;
  }

  location /swagger-ui {
    include proxy_params;
    proxy_pass http://correctexam-back:8080/swagger-ui;
  }

  location /v3/api-docs {
    include proxy_params;
    proxy_pass http://correctexam-back:8080/v3/api-docs;
  }

  location /auth {

```

(continues on next page)

(continued from previous page)

```
    include proxy_params;
    proxy_pass http://correctexam-back:8080/auth;
}

location /health {
    include proxy_params;
    proxy_pass http://correctexam-back:8080/health;
}

location / {
    root    /usr/share/nginx/html;
    index  index.html index.htm;
    try_files $uri $uri/ /index.html?$args;
}

#error_page 404          /404.html;

# redirect server error pages to the static page /50x.html
#
error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root    /usr/share/nginx/html;
}
}
```

nginx.conf

```
user  nginx;
worker_processes  auto;

error_log  /var/log/nginx/error.log notice;
pid        /var/run/nginx.pid;

events {
    worker_connections  1024;
}

http {
    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 1000s;
    types_hash_max_size 2048;

    include      /etc/nginx/mime.types;
    default_type application/octet-stream;
```

(continues on next page)

(continued from previous page)

```

log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                '$status $body_bytes_sent "$http_referer" '
                '"$http_user_agent" "$http_x_forwarded_for"';

access_log /var/log/nginx/access.log main;

sendfile      on;
#tcp_nopush   on;

keepalive_timeout 65;

gzip on;
    client_max_body_size 900M;
    client_body_buffer_size 900M;

include /etc/nginx/conf.d/*.conf;
}

```

2.3 Or Deploy the database and the backend on your own infrastructure and the frontend on a CDN

If you want to deploy the database and the backend infrastructure on your own infrastructure and deploy the frontend on the CDN. You must have to manage properly your CORS authorization within your CDN and within your backend. If you use github page public site, Pages allows CORS (access-control-allow-origin header is set to *). For the backend, you can use the quarkus properties `-Dquarkus.http.cors=true -Dquarkus.http.cors.origins=https://correctexam.github.io -Dquarkus.http.cors.methods=GET,PUT,POST,DELETE,PATCH,OPTIONS` to manage your cors. Please update the docker-compose descriptor accordingly.

To start your backend and your frontend, I will propose the skeleton of a docker-compose. Please update it for your needs. On top of that, you can configure bunkerized nginx to automatically setup your security and let's encrypt certificate. This docker compose automatically setup the database and the backend. You must update the `./resources/db/migration/sql` file that will be executed when the database will start. It populates the database with initial data.

```

version: '2'
services:
  correctexam-back
    image: barais/correctexam-back:manifest-amd64
    volumes:
# Path for
    - /tmp/files:/tmp/files:rw
    restart: always
    ports:
    - 8080:8080
# All quarkus configuration parameters (knobs) could be override through command line.
↳ You could also use different options to update the configuration parameters. https://
↳ quarkus.io/guides/config-reference
    command: ./application -Dquarkus.http.host=0.0.0.0 -Dquarkus.datasource.

```

(continues on next page)

(continued from previous page)

```
↪username=root -Dquarkus.datasource.password='' -Dquarkus.datasource.jdbc.
↪url=jdbc:mysql://correctexam-mysql:3306/correctexam?useUnicode=true&
↪characterEncoding=utf8&useSSL=false -Dquarkus.http.cors=true -Dquarkus.http.cors.
↪origins=https://correctexam.github.io -Dquarkus.http.cors.methods=GET,PUT,POST,DELETE,
↪PATCH,OPTIONS -Dquarkus.http.cors.headers=accept,origin,authorization,content-type,x-
↪requested-with -Dquarkus.http.cors.exposed-headers=Content-Disposition -Dquarkus.http.
↪cors.access-control-max-age=24H -Dquarkus.mailer.from=olivier.barais@univ-rennes1.fr -
↪Dquarkus.mailer.host=partage.univ-rennes1.fr -Dquarkus.mailer.port=587 -Dquarkus.
↪mailer.ssl=false -Dquarkus.mailer.username=olivier.barais@univ-rennes1.fr -Dquarkus.
↪mailer.password=TOCHANGE -Dhipster.mail.base-url=https://correctexam.github.io/
↪corrigeExamFront
correctexam-mysql:
  image: mysql:8.0.20
  volumes:
    - ../../resources/db/migration/:/docker-entrypoint-initdb.d
  environment:
    - MYSQL_USER=root
    - MYSQL_ALLOW_EMPTY_PASSWORD=yes
    - MYSQL_DATABASE=correctexam
  command: mysqld --lower_case_table_names=1 --skip-ssl --character_set_server=utf8mb4
↪--explicit_defaults_for_timestamp
  ports:
    - 3308:3306
```

Before building the frontend for your CDN, do not forget to update **webpack/environment.js** with your domain names.

BUILD AND DEPLOY ON RASPBERRY PI (ARM64)

3.1 Install support of cross compile on your machine

```
sudo apt-get install qemu binfmt-support qemu-user-static
docker run --rm --privileged multiarch/qemu-user-static --reset -p yes # This step will
↳ execute the registering scripts
docker run --rm -t arm64v8/ubuntu uname -m
```

3.2 Build

3.2.1 Build the backend

Create a docker file for quarkus to cross compile on your machine for arm64

```
FROM ghcr.io/graalvm/graalvm-ce:ol8-
↳ java11@sha256:dc9effae9a92d50e0a173f1cb8113409a4b6d7fb0c44fcf2195f0e03d6161bc5 AS build
RUN gu install native-image
WORKDIR /project
VOLUME ["/project"]
ENTRYPOINT ["native-image"]
```

Build the image

```
docker build -f src/main/docker/Dockerfile.build.aarch64 -t barais/quarkus-build-aarch64
↳ .
```

Next build your executable

```
./mvnw clean package -Pnative -Pprod -Dquarkus.package.type=native -DskipTests=true -
↳ Dquarkus.native.container-build=true -Dquarkus.native.builder-image=barais/quarkus-
↳ build-aarch64:latest
```

OR build it using quarkus cli

```
quarkus build --native --no-tests -Dquarkus.native.container-build=true -Dquarkus.native.
↳ builder-image=barais/quarkus-build-aarch64
```

When the compilation will over (could take a long time ;), you can build the final images with the binary.

Create the base image for your backend

```
FROM registry.access.redhat.com/ubi8/ubi-
↳minimal@sha256:c6592eb9cdd7ea7fa43beddf507ca2a8c2127f13ef66d49baea2fd28e37f62ba
WORKDIR /work/
RUN chown 1001 /work \
    && chmod "g+rwX" /work \
    && chown 1001:root /work
#COPY --chown=1001:root target/*-runner /work/application
# COPY --chown=1001:root ./src/main/resources/db/migration/ /work/migration
COPY target/*-runner /work/application
RUN chmod 775 /work
EXPOSE 8080
USER 1001
CMD ["/application", "-Dquarkus.http.host=0.0.0.0"]
```

```
docker build -f src/main/docker/Dockerfile.arm64 -t barais/correctexam-back:manifest-
↳arm64v8 --build-arg ARCH=arm64v8/ .
```

3.2.2 Build the frontend

Clone the frontend repository.

update webpack/environment.js with your domain name.

```
git clone https://github.com/correctexam/corrigeExamFront
cd corrigeExamFront
# update webpack/environment.js with your domain name
sudo docker build -f src/main/docker/Dockerfile.arm64 -t barais/correctexam-
↳front::manifest-arm64v8 --build-arg ARCH=arm64v8/ .
# OR using buildx
sudo docker buildx build -f src/main/docker/Dockerfile --push --platform linux/arm64,
↳linux/amd64 --tag barais/correctexam-front .
```

You will obtain a nginx with only the js, html and js. You have to mount the configuration if you want to manage proxy to the backend routes. I would prefer to use bunkerized nginx for the routing.

3.3 Deploy on your raspberry 4

You can push your built image on dockerhub (update docker image within the docker compose) and just deploy the docker compose on your own raspberry with the nginx configuration files.

```
version: '2'
services:
  correctexam-back:
    image: barais/correctexam-back::manifest-arm64v8
    volumes:
# Path for
    - /tmp/files:/tmp/files:rw
    restart: always
    ports:
    - 8080:8080
```

(continues on next page)

(continued from previous page)

```
# All quarkus configuration parameters (knobs) could be override through command line.
↳ You could also use different options to update the configuration parameters. https://quarkus.io/guides/config-reference
  command: ./application -Dquarkus.http.host=0.0.0.0 -Dquarkus.datasource.
↳ username=root -Dquarkus.datasource.password='' -Dquarkus.datasource.jdbc.
↳ url=jdbc:mysql://correctexam-mysql:3306/correctexam?useUnicode=true&
↳ characterEncoding=utf8&useSSL=false -Dquarkus.http.cors=true -Dquarkus.http.cors.
↳ origins=https://correctexam.github.io -Dquarkus.http.cors.methods=GET,PUT,POST,DELETE,
↳ PATCH,OPTIONS -Dquarkus.http.cors.headers=accept,origin,authorization,content-type,x-
↳ requested-with -Dquarkus.http.cors.exposed-headers=Content-Disposition -Dquarkus.http.
↳ cors.access-control-max-age=24H -Dquarkus.mailer.from=olivier.barais@univ-rennes1.fr -
↳ Dquarkus.mailer.host=partage.univ-rennes1.fr -Dquarkus.mailer.port=587 -Dquarkus.
↳ mailer.ssl=false -Dquarkus.mailer.username=olivier.barais@univ-rennes1.fr -Dquarkus.
↳ mailer.password=TOCHANGE -Dhipster.mail.base-url=https://correctexam.github.io/
↳ corrigeExamFront
correctexam-mysql:
  image: mysql:8.0.20
  volumes:
    - ../../resources/db/migration/:/docker-entrypoint-initdb.d
  environment:
    - MYSQL_USER=root
    - MYSQL_ALLOW_EMPTY_PASSWORD=yes
    - MYSQL_DATABASE=correctexam
  command: mysqld --lower_case_table_names=1 --skip-ssl --character_set_server=utf8mb4
↳ --explicit_defaults_for_timestamp
#   ports:
#     - 3308:3306
front:
  image: barais/correctexam-front:manifest-arm64v8
#   ports:
#     - 90:80
  volumes:
    - ./exampleconf/exam.conf:/etc/nginx/conf.d/exam.conf
    - ./exampleconf/nginx.conf:/etc/nginx/nginx.conf:ro
```

exam.conf and **nginx.conf** could be something like that (you have to update the server name)

exam.conf

```
server {
  listen      80;
  listen     [::]:80;
  # server name to change based on your own domain name for your front
  server_name correctexam.barais.fr;

  #charset koi8-r;
  #access_log /var/log/nginx/host.access.log main;

  location /api {
    proxy_pass http://correctexam-back:8080/api;
    proxy_set_header Host $http_host;
  }
}
```

(continues on next page)

```
location /api {
    include proxy_params;
    proxy_pass http://correctexam-back:8080/api;
}

location /management {
    include proxy_params;
    proxy_pass http://correctexam-back:8080/management;
}

location /swagger-ui {
    include proxy_params;
    proxy_pass http://correctexam-back:8080/swagger-ui;
}

location /v3/api-docs {
    include proxy_params;
    proxy_pass http://correctexam-back:8080/v3/api-docs;
}

location /auth {
    include proxy_params;
    proxy_pass http://correctexam-back:8080/auth;
}

location /health {
    include proxy_params;
    proxy_pass http://correctexam-back:8080/health;
}

location / {
    root    /usr/share/nginx/html;
    index  index.html index.htm;
    try_files $uri $uri/ /index.html?$args;
}

#error_page 404          /404.html;

# redirect server error pages to the static page /50x.html
#
error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root    /usr/share/nginx/html;
}
}
```

nginx.conf

```
user nginx;
worker_processes auto;

error_log /var/log/nginx/error.log notice;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 1000s;
    types_hash_max_size 2048;

    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;

    sendfile on;
    #tcp_nopush on;

    keepalive_timeout 65;

    gzip on;
    client_max_body_size 900M;
    client_body_buffer_size 900M;

    include /etc/nginx/conf.d/*.conf;
}
```


CREATE A RELEASE ON DOCKER HUB

Next for the front you can use dockerX or create your docker image for the different targeted architecture.

```
docker manifest create \  
barais/correctexam-front:manifest-v1 \  
--amend barais/correctexam-front:manifest-amd64 \  
--amend barais/correctexam-front:manifest-arm64v8 \  
docker manifest push barais/correctexam-front:manifest-v1
```

For the back, create your docker image for the different targeted architecture.

```
docker manifest create \  
barais/correctexam-back:manifest-v1 \  
--amend barais/correctexam-back:manifest-amd64 \  
--amend barais/correctexam-back:manifest-arm64v8 \  
docker manifest push barais/correctexam-back:manifest-v1
```