Numbas LTI provider Documentation

Release v3.0-development

Numbas

Jul 07, 2022

Introduction

This is a Basic LTI 1.1 tool provider, to run Numbas exams in any LTI-compatible virtual learning environment.

The tool handles attempt data, and as well as offering CSV exports of student scores can report scores back to the host VLE's gradebook.

CHAPTER 1

Contents

1.1 How it works

- *Install the software* on your own server.
- *Create an LTI consumer key* and add the LTI tool to your VLE using the details provided.
- Add a link to the tool from your course's page on the VLE and upload a Numbas exam package.
- Students click on the link to attempt the exam. Their answers and scores are saved as soon as they are submitted. Students can leave the exam and resume it later by clicking on the link again.
- Instructors who click on the link can *review students' attempts*, *download a summary of scores*, or *report scores back to the VLE's gradebook*.

1.2 Data handling

This page aims to answer some common queries about how data is handled by the Numbas LTI tool.

The Numbas LTI tool provider runs entirely on your own server. All data is saved on your server, and nothing is sent back to Newcastle University.

You are responsible for ensuring the security of data held by the Numbas LTI tool and for managing the data lifecycle.

When a *resource* on the LTI provider is launched, the client's device sends a request containing some data generated by the *tool consumer*, following the Basic LTI standard. This information is cryptographically signed using the OAuth protocol. It contains at least a unique identifier for the link and a unique identifier for the user opening the link. This identifier might correspond to a username on the consumer.

The LTI tool can send scores for students' attempts at Numbas *resources* back to the *tool consumer*, using the Basic LTI Outcomes Service. Data is sent using a request signed using OAuth, to an address provided by the tool consumer.

If you have set up an *editor link*, the LTI tool will occasionally request a list of available exams from the linked editor. This request is made anonymously, and no personally identifying information is sent.

1.2.1 User roles

Administrator

A user with ‘administrator’ or ‘superuser’ privileges can see all data held by the LTI tool provider. They can manage consumers through the management interface, and access all resources without making an LTI launch.

Instructor

When a user opens a resource through an LTI launch which specifies the ‘Instructor’ role, they are shown the management dashboard for that resource.

They can see all student attempt data relating to the resource.

They can download summary data about student attempts, including identifiers and scores.

Student

When a user opens a resource through an LTI launch which specifies the ‘Student’ role, they are shown their own attempts at that resource. They can not see any other student’s attempts.

1.3 Getting started

This document will lead you through the process of setting up an instance of the Numbas LTI tool provider on your own server, and adding a Numbas activity to a course in your VLE.

1.3.1 What do I need in order to use this?

You must run your own instance of this tool - as well as using a fair amount of server resources, we don’t want to keep other people’s student data!

To install the tool, you need:

- A dedicated server to install the software on, which will communicate with your virtual learning environment. At Newcastle, we use servers running Ubuntu, though anything that can run Python will do. As a rough guide, you’ll need around 0.5MB of disk space per student attempt.
- The ability to add a link to an LTI tool to your virtual learning environment. In Blackboard and Moodle, only administrators can do this, but it only needs to be done once.

At most institutions, this will require the help of your IT team.

If you’re unsure whether you can use the LTI tool provider, or want help setting it up, email the Numbas team.

1.3.2 Install the software

Follow the *installation instructions* to set up an instance of the Numbas LTI tool provider on your own server.

1.3.3 Link your VLE

Create a consumer in the tool provider's admin interface, and connect to it from your VLE.

This may require support from the administrator of your VLE.

1.3.4 Link a Numbas editor

Note: This step is optional.

You can *link to an instance of the Numbas editor* to make a selection of exams available directly from the LTI tool, instead of having to download a Numbas exam package to your own computer first.

1.3.5 Add a Numbas activity to your course

Add a link from your course's page in your VLE to the Numbas tool provider.

Download a Numbas exam package from the Numbas editor and upload it to the newly-created resource.

Students who click on the link will be shown the exam, and their answers and scores will be saved to the tool provider.

1.3.6 Get scores for the activity

You can *download a .CSV file* of the students' scores or *report their scores back to the VLE's gradebook*.

1.4 Installation

We have written installation instructions for a selection of common platforms.

The Numbas LTI tool provider can be installed on any platform that can run Python or Docker. The simplest installation method is *Docker*.

If you're installing on a platform not listed below, follow the *Ubuntu installation instructions* and make changes as necessary. Contact the Numbas team if you need help.

1.4.1 Installation with Docker

You will need:

- Access to a server where you can run Docker and Docker Compose. A virtual machine is fine.
- A domain name pointed at the server you're going to use. A subdomain (e.g. `numbas-lti.youruniversity.edu`) is fine.
- An SSL certificate: LTI content must be served over HTTPS. These can be obtained easily and for free from Let's Encrypt.

These instructions will run the Numbas LTI provider inside a Docker container, along with containers for the services it requires.

Setup

Obtain the Docker compose recipe and its associated files either by downloading the latest release from GitHub, or by cloning the git repository:

```
git clone https://github.com/numbas/numbas-lti-provider-docker.git
```

Copy the file `settings.env.dist` to `settings.env` and write your own values for each of the variables inside.

Run the `get_secret_key` script to generate a value for the `SECRET_KEY` environment variable, and put that in `settings.env`:

```
docker-compose run --rm numbas-setup python ./get_secret_key
```

Obtain an SSL certificate and key for the domain you will access the Numbas LTI provider from. Copy the key to `files/ssl/numbas-lti.key` and the certificate to `files/ssl/numbas-lti.pem`.

Run the installation script, to set up the database and create the superuser account:

```
docker-compose run --rm numbas-setup python ./install
```

The LTI provider is ready to start.

Starting

Run the following command:

```
docker-compose up --scale daphne=4 --scale huey=2
```

You can customise the number of each of the kinds of process by changing the numbers in the –*scale* arguments. You'll have to establish how many of each you need by experimentation. The *daphne* process handles web requests; you will need more if you have lots of simultaneous connections. The *huey* process runs asynchronous tasks; you will need more if you find that tasks such as reporting scores or generating report files take a very long time.

Stopping

Stop the containers with:

```
docker-compose down
```

This does not delete any permanent data such as the database, settings file or uploaded files.

Upgrading

First, update the files in the Docker repository. If you used git, run:

```
git pull
```

Upgrade to a new version of the Numbas LTI provider with the following commands:

```
docker-compose build --no-cache numbas-setup
docker-compose build --no-cache daphne
docker-compose build --no-cache huey
docker-compose run --rm numbas-setup python ./install
docker-compose restart
```

Occasionally there are other changes that must be made; check the *upgrade instructions* for the version you are upgrading to.

Running in the cloud

Docker Compose files can also be used to deploy to the cloud. See the following documents for more information about deploying Docker to the cloud:

- Compose for Amazon ECS
- Compose for Microsoft ACI

1.4.2 Installation on Ubuntu

You will need:

- Administrator access to a server with at least 4GB of RAM, connected to the internet. A virtual machine is fine.
- A domain name pointed at the server you're going to use. A subdomain (e.g. `numbas-lti.youruniversity.edu`) is fine.
- An SSL certificate: LTI content must be served over HTTPS. These can be obtained easily and for free from Let's Encrypt.

These instructions will take a fresh machine running Ubuntu 16.04 or newer and set up the Numbas LTI tool provider to run through NGINX or Apache. On different operating systems or with different web servers, the process will be different. There are alternate instructions for *installation on RedHat Enterprise Linux 7*.

The Numbas LTI provider is a Django app. See the Django documentation for configuration options not described here.

We will set up:

- The Numbas LTI provider Django app running inside a virtual Python environment, isolated from the system's Python environment.
- A PostgreSQL database for the LTI provider to use. You can use any database supported by Django; see the list of supported databases in the Django documentation.
- Supervisord will ensure the app is always running.
- The NGINX webserver as a reverse proxy to serve the LTI provider to the outside world. If necessary, you can use Apache as a reverse proxy (see the guide in the Apache documentation), but it can not handle as many simultaneous connections.

Set up the environment

First, install packages, set up users, and create the required paths (you can save this script as a file and run it as root - it doesn't need any user input):

```
#!/usr/bin/bash

# set up user group
adduser --disabled-password numbas_lti
adduser www-data numbas_lti

# install packages
apt update
```

(continues on next page)

(continued from previous page)

```
apt install \
    git redis-server postgresql postgresql-server-dev-all \
    libxml2-dev libxslt1-dev python-dev lib32z1-dev python3-pip supervisor
pip3 install virtualenv

# get the numbas-lti-provider code
git clone https://github.com/numbas/numbas-lti-provider.git /srv/numbas-lti-provider
cd /srv/numbas-lti-provider
git checkout v3_STABLE
chown -R numbas_lti:numbas_lti /srv/numbas-lti-provider

# create media and static file directories
mkdir /srv/numbas-lti-media
mkdir /srv/numbas-lti-static
chown -R numbas_lti:numbas_lti /srv/numbas-lti-media
chown -R www-data:www-data /srv/numbas-lti-static
chmod -R 770 /srv/numbas-lti-media /srv/numbas-lti-static

# create the virtualenv for the python modules
virtualenv -p python3 /opt/numbas_lti_python
chown -R numbas_lti:numbas_lti /opt/numbas_lti_python
chmod -R 770 /opt/numbas_lti_python

# install python modules
cd /srv/numbas-lti-provider
source /opt/numbas_lti_python/bin/activate
pip install -r requirements.txt
pip install channels_redis==3.3.0 psycopg2==2.8.6 redis==3.5.3
```

Next, create a database and set a password to access it (replace `$password` with your chosen password in the following script):

```
sudo -u postgres psql -c "CREATE USER numbas_lti WITH ENCRYPTED PASSWORD '$password'␣
↪CREATEDB;"
sudo -u postgres createdb -U numbas_lti numbas_lti -h localhost
```

Configuring the Numbas LTI provider

Run:

```
cd /srv/numbas-lti-provider
source /opt/numbas_lti_python/bin/activate
python first_setup.py
```

This script will ask a few questions, and configure the Numbas LTI provider accordingly. It will set up the database, and create an admin user account which you will use to manage the LTI provider through its web interface.

Note: The first question that the setup script asks is "Is this installation for development?". The settings for development mode are not compatible with serving the LTI provider to external clients.

Only answer 'yes' to this question if the installation is for the purpose of making changes to the LTI provider's code. For all other purposes, answer 'no'.

Once you've run this script, the last remaining steps are to start the app, and then set up a webserver to expose it to the outside world.

Configure supervisord

Supervisord ensures that the Numbas LTI provider app is always running.

Save the following as /etc/supervisor/conf.d/numbas_lti.conf:

```
[program:numbas_lti_daphne]
command=/opt/numbas_lti_python/bin/daphne numbasltiprovider.asgi:application --port 87
↪%(process_num)02d --bind 0.0.0.0 -v 2
directory=/srv/numbas-lti-provider/
user=www-data
autostart=true
autorestart=true
stopasgroup=true
environment=DJANGO_SETTINGS_MODULE=numbasltiprovider.settings
numprocs=4
process_name=%(program_name)s_%(process_num)02d
stderr_logfile=/var/log/supervisor/numbas_lti_daphne_stderr.log
stdout_logfile=/var/log/supervisor/numbas_lti_daphne_stdout.log

[program:numbas_lti_huey]
command=/opt/numbas_lti_python/bin/python /srv/numbas-lti-provider/manage.py run_huey␣
↪-w 8
directory=/srv/numbas-lti-provider/
user=www-data
autostart=true
autorestart=true
redirect_stderr=True
stopasgroup=true
environment=DJANGO_SETTINGS_MODULE="numbasltiprovider.settings"
numprocs=1
process_name=%(program_name)s_%(process_num)02d
stderr_logfile=/var/log/supervisor/numbas_lti_huey_stderr.log
stdout_logfile=/var/log/supervisor/numbas_lti_huey_stdout.log

[group:numbas_lti]
programs=numbas_lti_daphne,numbas_lti_huey
priority=999
```

Note: If your server must use a proxy to make HTTP or HTTPS requests, you should set environment variables HTTP_PROXY and HTTPS_PROXY in the supervisor configuration. Add them to the lines starting environment=, for example:

```
environment=DJANGO_SETTINGS_MODULE="numbasltiprovider.settings",HTTP_PROXY=http://web.
↪proxy:4321,HTTPS_PROXY=http://web.proxy:4321
```

Once you've set this up, run:

```
systemctl restart supervisor
```

Supervisord will start the Numbas LTI provider, and restart it automatically if it ever crashes.

Set up a webserver

We have instructions for two webservers: *NGINX* and *Apache*.

With NGINX

NGINX is a high performance webserver, ideal for use as a reverse proxy. It is the recommended option for the Numbas LTI provider.

Install NGINX:

```
apt install nginx
```

Overwrite `/etc/nginx/sites-available/default` with the following:

```
upstream backend_hosts {
 server 0.0.0.0:8700;
 server 0.0.0.0:8701;
 server 0.0.0.0:8702;
 server 0.0.0.0:8703;
}

server {
    listen 443;
    client_max_body_size 20M;

    ssl on;
    ssl_certificate /etc/ssl/numbas-lti.pem;
    ssl_certificate_key /etc/ssl/numbas-lti.key;

    error_page 502 /502.html;
    location = /502.html {
      root /srv/www/server-error;
    }

    location /static {
        alias /srv/numbas-lti-static;
    }

    location /media {
        alias /srv/numbas-lti-media;
    }

    location / {
        proxy_pass http://backend_hosts;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_buffering off;
        proxy_redirect     off;
        proxy_set_header   Host $host;
        proxy_set_header   X-Real-IP $remote_addr;
        proxy_set_header   X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header   X-Forwarded-Host $server_name;
        proxy_set_header   X-Scheme https;
        proxy_set_header   X-Forwarded-Proto https;
        proxy_read_timeout 600s;
```

(continues on next page)

(continued from previous page)

```
    }

}
```

Set the `ssl_certificate` and `ssl_certificate_key` lines to the paths to your SSL certificate and key files. If you're using **`certbot`**, it will add those lines for you.

You should put something in `/srv/www/server-error/502.html`, to be shown when there's a server error. This can happen if the Numbas LTI provider isn't running, or otherwise fails to communicate with NGINX.

With Apache

Apache is a very commonly-used webserver.

> **Warning:** While it can be used as a reverse proxy for the Numbas LTI provider, it's not great at dealing with the many simultaneous connections that the LTI provider requires. Apache will start having trouble at around 100 simultaneous connections. In some circumstances, Apache might be your only option, so the instructions are provided as a reference.

Install required packages:

```
apt install apache2
a2enmod ssl proxy proxy_wstunnel proxy_http proxy_connect headers rewrite
```

Overwrite `/etc/apache2/sites-available/000-default.conf` with the following:

```
<VirtualHost *:443>
  SSLEngine on
  SSLProxyEngine on
  SSLCertificateFile /etc/apache2/ssl/certs/numbas_lti.crt
  SSLCertificateKeyFile /etc/apache2/ssl/private/numbas_lti.key

  ProxyPreserveHost On
  ProxyRequests Off
  ProxyPass /static !
  Alias "/static" "/srv/numbas-lti-static"
  ProxyPass /media !
  Alias "/media" "/srv/numbas-lti-media"
  ProxyPass "/websocket" "ws://0.0.0.0:8700/websocket"
  ProxyPassReverse "/websocket" "ws://0.0.0.0:8700/websocket"
  ProxyPass / http://0.0.0.0:8700/
  ProxyPassReverse / http://0.0.0.0:8700/

  RequestHeader set X-Scheme "https"
  RequestHeader set X-Forwarded-Proto "https"

  <Directory "/srv/numbas-lti-static">
    AllowOverride None
    Options FollowSymLinks
    Require all granted
  </Directory>

  <Directory "/srv/numbas-lti-media">
```

(continues on next page)

(continued from previous page)

```
    AllowOverride None
    Options FollowSymLinks
    Require all granted
  </Directory>

  ErrorLog ${APACHE_LOG_DIR}/numbas_lti.error.log
  CustomLog ${APACHE_LOG_DIR}/numbas_lti.access.log combined
</VirtualHost>
```

Set the `SSLCertificateFile` and `SSLCertificateKeyFile` lines to the paths to your SSL certificate and key files. If you're using certbot, it will add those lines for you.

Obtain an SSL certificate

An SSL certificate allows your server to communicate with browsers securely.

The easiest way of obtaining an SSL certificate is with certbot, from the EFF. It's a command-line tool which automatically acquires certificates from Let's Encrypt for any domains you're serving. Follow the instructions on the certbot site, after setting up your web server, to obtain a certificate.

These certificates don't last very long, and need to be renewed. You can do this automatically by running `certbot renew` as a cron job; put the following in `/etc/cron.daily/renew-certbot`:

```
#!/bin/sh
certbot renew
```

Make sure that `/etc/cron.daily/renew-certbot` is executable by the root user:

```
chmod +x /etc/cron.daily/renew-certbot
```

If you have no other way of obtaining a certificate, you can create a self-signed certificate which will produce a security warning in web browsers.

Ensure outcome reporting works

In order to report scores back to the *tool consumer*, the Numbas LTI provider must make an HTTPS request to an address provided by the consumer. Normally, this is on the same domain as the consumer.

Ensure that the machine on which the LTI provider is running can make HTTPS requests to the consumer - if you're working in a testing environment, you may need to configure the consumer's server to allow connections on port 443 from the provider's IP address.

Updating the software

You should keep the software up-to-date with any bugfixes or new features.

Run the following:

```
cd /srv/numbas-lti-provider
git pull origin
source /opt/numbas_lti_python/bin/activate
pip install -r requirements.txt
python manage.py migrate
```

(continues on next page)

(continued from previous page)

```
python manage.py collectstatic --noinput
supervisorctl restart numbas_lti:
```

Ready to use

Once you've got everything running, the LTI provider will be available to use, at the domain name you configured.

Open the site in a web browser and log in using the admin account credentials you set up earlier.

If you encounter any problems, see the *Troubleshooting* page.

The next step is to add an LTI consumer key so that your VLE can connect to the LTI provider.

1.4.3 Installation on RedHat Enterprise Linux 7

Note: These instructions were tested using a virtual machine running RedHat Enterprise Linux 7.8 on 2020-08-04. Software changes continually, so some steps might not work as described.

If you have any trouble, please file an issue on GitHub or email the Numbas team.

You will need:

- Administrator access to a server with at least 4GB of RAM, connected to the internet. A virtual machine is fine.
- A domain name pointed at the server you're going to use. A subdomain (e.g. `numbas-lti.youruniversity.edu`) is fine.
- An SSL certificate: LTI content must be served over HTTPS. These can be obtained easily and for free from Let's Encrypt.

These instructions will take a fresh machine using RHEL 7 and set up the Numbas LTI tool provider to run through NGINX. On different operating systems or with different web servers, the process will be different. There are alternate instructions for *installation on Ubuntu*.

The Numbas LTI provider is a Django app. See the Django documentation for configuration options not described here.

We will set up:

- The Numbas LTI provider Django app running inside a virtual Python environment, isolated from the system's Python environment.
- A PostgreSQL database for the LTI provider to use. You can use any database supported by Django; see the list of supported databases in the Django documentation.
- Supervisord will ensure the app is always running.
- The NGINX webserver as a reverse proxy to serve the LTI provider to the outside world. If necessary, you can use Apache as a reverse proxy (see the guide in the Apache documentation), but it can not handle as many simultaneous connections.

Set up the environment

Make sure that your machine has an active subscription using **`subscription-manager`**.

You may need to enable some extra repositories:

```
subscription-manager repos --enable "rhel-*-optional-rpms" --enable "rhel-*-extras-
↪rpms"  --enable "rhel-ha-for-rhel-*-server-rpms"
```

In order to install NGINX, create the `/etc/yum.repos.d/nginx.repo` with the following contents:

```
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/rhel/7/$basearch/
gpgcheck=0
enabled=1
```

Set up a user account and install packages, and create the required paths. Run the following commands as root:

```
#!/bin/bash
useradd numbas_lti
yum install https://download.postgresql.org/pub/repos/yum/10/redhat/rhel-7-x86_64/
↪pgdg-redhat-repo-latest.noarch.rpm
yum install @development
yum install postgresql96-server postgresql96 \
    python36 python36-pip python36-devel \
    python3-virtualenv \
    redis supervisor nginx git

systemctl enable redis
systemctl start redis

# create media and static file directories
mkdir /srv/numbas-lti-media
mkdir /srv/numbas-lti-static
chown -R numbas_lti:numbas_lti /srv/numbas-lti-media
chown -R nginx:nginx /srv/numbas-lti-static
chmod -R 770 /srv/numbas-lti-media /srv/numbas-lti-static

# get the numbas-lti-provider code
git clone https://github.com/numbas/numbas-lti-provider.git /srv/numbas-lti-provider
cd /srv/numbas-lti-provider
git checkout v3_STABLE
chown -R numbas_lti:numbas_lti /srv/numbas-lti-provider

# create the virtualenv for the python modules
virtualenv -p python3 /opt/numbas_lti_python
chown -R numbas_lti:numbas_lti /opt/numbas_lti_python
chmod -R 770 /opt/numbas_lti_python

# install python modules
cd /srv/numbas-lti-provider
source /opt/numbas_lti_python/bin/activate
pip install -r requirements.txt
pip install channels_redis==3.3.0 psycopg2==2.8.6 redis==3.5.3
```

Change PostgreSQL to use password authentication: edit `/var/lib/psql/9.6/data/pg_hba.conf`, and change:

```
host    all             all             127.0.0.1/32            ident
```

to:

```
host    all             all             127.0.0.1/32            md5
```

Now restart PostgreSQL and create a database:

```
systemctl restart postgresql-9.6
sudo -u postgres psql -c "CREATE USER numbas_lti WITH ENCRYPTED PASSWORD '$password' CREATEDB;"
sudo -u postgres createdb -U numbas_lti numbas_lti -h localhost
```

Configuring the Numbas LTI provider

Run:

```
cd /srv/numbas-lti-provider
source /opt/numbas_lti_python/bin/activate
python first_setup.py
```

This script will ask a few questions, and configure the Numbas LTI provider accordingly. It will set up the database, and create an admin user account which you will use to manage the LTI provider through its web interface.

Note: The first question that the setup script asks is “Is this installation for development?”. The settings for development mode are not compatible with serving the LTI provider to external clients.

Only answer ‘yes’ to this question if the installation is for the purpose of making changes to the LTI provider’s code. For all other purposes, answer ‘no’.

Once you’ve run this script, the last remaining steps are to start the app, and then set up a webserver to expose it to the outside world.

Configure supervisord

Supervisord ensures that the Numbas LTI provider app is always running.

Save the following as `/etc/supervisord.d/numbas_lti.ini`:

```
[program:numbas_lti_daphne]
command=/opt/numbas_lti_python/bin/daphne numbasltiprovider.asgi:application --port 87%(process_num)02d --bind 0.0.0.0 -v 2
directory=/srv/numbas-lti-provider/
user=www-data
autostart=true
autorestart=true
stopasgroup=true
environment=DJANGO_SETTINGS_MODULE=numbasltiprovider.settings
numprocs=4
process_name=%(program_name)s_%(process_num)02d
stderr_logfile=/var/log/supervisor/numbas_lti_daphne_stderr.log
stdout_logfile=/var/log/supervisor/numbas_lti_daphne_stdout.log

[program:numbas_lti_huey]
command=/opt/numbas_lti_python/bin/python /srv/numbas-lti-provider/manage.py run_huey -w 8
directory=/srv/numbas-lti-provider/
```

(continues on next page)

(continued from previous page)

```
user=www-data
autostart=true
autorestart=true
redirect_stderr=True
stopasgroup=true
environment=DJANGO_SETTINGS_MODULE="numbasltiprovider.settings"
numprocs=1
process_name=%(program_name)s_%(process_num)02d
stderr_logfile=/var/log/supervisor/numbas_lti_huey_stderr.log
stdout_logfile=/var/log/supervisor/numbas_lti_huey_stdout.log

[group:numbas_lti]
programs=numbas_lti_daphne,numbas_lti_huey
priority=999
```

Note: If your server must use a proxy to make HTTP or HTTPS requests, you should set environment variables `HTTP_PROXY` and `HTTPS_PROXY` in the supervisor configuration. Add them to the lines starting `environment=`, for example:

```
environment=DJANGO_SETTINGS_MODULE="numbasltiprovider.settings",HTTP_PROXY=http://web.proxy:4321,HTTPS_PROXY=http://web.proxy:4321
```

Once you've set this up, run:

```
systemctl restart supervisord
```

Supervisord will start the Numbas LTI provider, and restart it automatically if it ever crashes.

Set up the NGINX webserver

NGINX is a high performance webserver, ideal for use as a reverse proxy. It is the recommended option for the Numbas LTI provider.

Add the *nginx* user to the *numbas_lti* group:

```
usermod -a -G numbas_lti nginx
```

Overwrite `/etc/nginx/conf.d/default.conf` with the following:

```
upstream backend_hosts {
 server 0.0.0.0:8700;
 server 0.0.0.0:8701;
 server 0.0.0.0:8702;
 server 0.0.0.0:8703;
}

server {
    listen 443;
    client_max_body_size 20M;

    ssl on;
    ssl_certificate /etc/ssl/numbas-lti.pem;
    ssl_certificate_key /etc/ssl/numbas-lti.key;
```

(continues on next page)

(continued from previous page)

```
    error_page 502 /502.html;
    location = /502.html {
      root /srv/www/server-error;
    }

    location /static {
        alias /srv/numbas-lti-static;
    }

    location /media {
        alias /srv/numbas-lti-media;
    }

    location / {
        proxy_pass http://backend_hosts;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_buffering off;
        proxy_redirect     off;
        proxy_set_header   Host $host;
        proxy_set_header   X-Real-IP $remote_addr;
        proxy_set_header   X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header   X-Forwarded-Host $server_name;
        proxy_set_header   X-Scheme https;
        proxy_set_header   X-Forwarded-Proto https;
        proxy_read_timeout 600s;
    }

}
```

Set the `ssl_certificate` and `ssl_certificate_key` lines to the paths to your SSL certificate and key files. If you're using **certbot**, it will add those lines for you.

You should put something in `/srv/www/server-error/502.html`, to be shown when there's a server error. This can happen if the Numbas LTI provider isn't running, or otherwise fails to communicate with NGINX.

Finally, open the firewall to allow web traffic:

```
setsebool -P httpd_can_network_connect 1
firewall-cmd --permanent --zone=public --add-service=http
firewall-cmd --permanent --zone=public --add-service=https
firewall-cmd --reload
setenforce permissive
systemctl start nginx
```

Obtain an SSL certificate

An SSL certificate allows your server to communicate with browsers securely.

The easiest way of obtaining an SSL certificate is with certbot, from the EFF. It's a command-line tool which automatically acquires certificates from Let's Encrypt for any domains you're serving. Follow the instructions on the certbot site, after setting up your web server, to obtain a certificate.

These certificates don't last very long, and need to be renewed. You can do this automatically by running `certbot renew` as a cron job; put the following in `/etc/cron.daily/renew-certbot`:

```
#!/bin/sh
certbot renew
```

Make sure that `/etc/cron.daily/renew-certbot` is executable by the root user:

```
chmod +x /etc/cron.daily/renew-certbot
```

If you have no other way of obtaining a certificate, you can create a self-signed certificate which will produce a security warning in web browsers.

Ensure outcome reporting works

In order to report scores back to the *tool consumer*, the Numbas LTI provider must make an HTTPS request to an address provided by the consumer. Normally, this is on the same domain as the consumer.

Ensure that the machine on which the LTI provider is running can make HTTPS requests to the consumer - if you're working in a testing environment, you may need to configure the consumer's server to allow connections on port 443 from the provider's IP address.

Updating the software

You should keep the software up-to-date with any bugfixes or new features.

Run the following:

```
cd /srv/numbas-lti-provider
git pull origin
source /opt/numbas_lti_python/bin/activate
pip install -r requirements.txt
python manage.py migrate
python manage.py collectstatic --noinput
supervisorctl restart numbas_lti:
```

Ready to use

Once you've got everything running, the LTI provider will be available to use, at the domain name you configured.

Open the site in a web browser and log in using the admin account credentials you set up earlier.

If you encounter any problems, see the *Troubleshooting* page.

The next step is to add an LTI consumer key so that your VLE can connect to the LTI provider.

1.4.4 Settings

The settings for the server are configured in the file `numbasltiprovider/settings.py`.

The `first_setup.py` script creates this file on installation. You might want to change settings if you have a nonstandard environment, or after upgrading.

The Numbas LTI provider is a Django app, so many of the settings come from Django. See the Django documentation for information on these.

This page describes the available settings which are unique to the Numbas LTI provider.

`INSTANCE_NAME`

The name of this instance of the LTI provider. You could use the name of your institution, or the name of the learning environment.

This is shown in the footer of each page.

`LTI_INSTRUCTOR_ROLES`

A list of LTI role names which grant the user 'instructor' status if they're present.

See the role vocabularies in the LTI spec.

`SUPPORT_NAME`

The name of your support contact. This will be shown to students in case of errors.

`SUPPORT_URL`

A URL to direct students to when there is an error. This could be a web address, a `mailto:` URI, or `None` if you don't want to display a link.

`EMAIL_COMPLETION_RECEIPTS`

If this is `True`, then the *Email attempt receipts to students on completion?* option will be available for each resource. Note that receipts are only sent for attempts at resources where the option is turned on by the instructor.

If this is `False`, then no completion receipts will be sent for any resource.

`REQUEST_TIMEOUT`

The number of seconds to wait when making an HTTP request to another service before giving up.

The following actions involve making HTTP requests to other services:

- Reporting scores back to the LTI consumer.
- Fetching updated exam packages from the Numbas editor
- Updating editor links.

`REPORT_FILE_EXPIRY_DAYS`

The number of days after which *report files* should be deleted.

`HELP_URL`

The address of the documentation for the LTI tool.

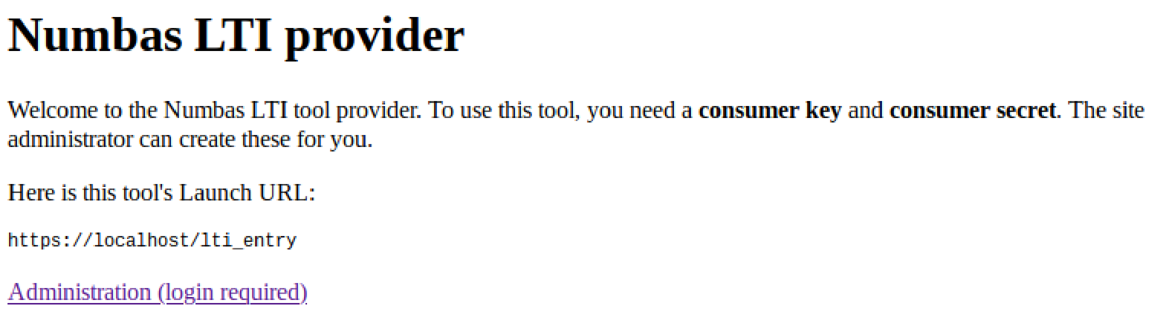
If this is not set, then the default of `https://docs.numbas.org.uk/lti/en/<version>` is used.

1.4.5 Troubleshooting

This page collects a few common problems encountered after installing the Numbas LTI provider.

If the problem you're having doesn't have a solution on this page, please file an issue on GitHub or email the Numbas team.

JavaScript and stylesheets don't load



Numbas LTI provider

Welcome to the Numbas LTI tool provider. To use this tool, you need a **consumer key** and **consumer secret**. The site administrator can create these for you.

Here is this tool's Launch URL:

```
https://localhost/lti_entry
```

Administration (login required)

Fig. 1: The server welcome page has loaded without any stylesheets.

If the page looks like the above screenshot, with no stying applied to the text, it's possible that there's a problem serving static files.

Try the following:

- Collect the static files:

```
cd /srv/numbas-lti-provider
source /opt/numbas_lti_python/bin/activate
python manage.py collectstatic --noinput
```

- Make sure the server can read the static files. On Ubuntu the web server runs as the user `www-data`, while on RHEL it's `nginx`:

```
chown -R www-data:www-data /srv/numbas-lti-static
```

- On a system with SELinux, you may need to set it to premissive mode:

```
setenforce Permissive
```

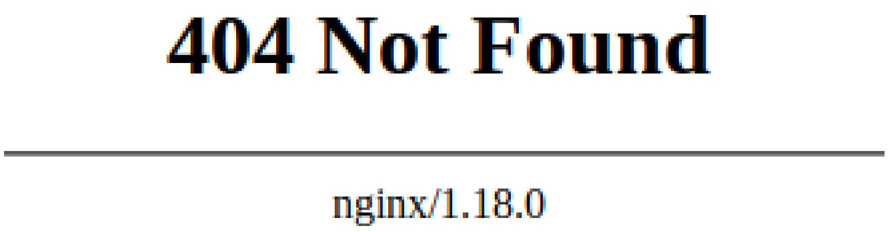
- If none of that works, open the browser's developer tools by pressing `F12` and look in the *Network* tab for more information on any errors.

Generic NGINX 404 error page

If you haven't set up a 502 error page (at :file:/srv/www/server-error/502.html‘ if you're following our instructions), it's likely that NGINX is really saying that it can't find a page to tell you about an error.

Have a look at the NGINX log, in `/var/log/nginx/error.log`.

If you see something like "no live upstreams" or "Connection refused while connecting to upstream", check that the Numbas LTI **`daphne`** and **`worker`** processes are running:



404 Not Found

nginx/1.18.0

Fig. 2: Nginx's "Not found" error page.

```
supervisorctl status
```

You should see something like the following:

```
numbas_lti:numbas_lti_daphne_00    RUNNING   pid 1999, uptime 0:37:48
numbas_lti:numbas_lti_daphne_01    RUNNING   pid 2000, uptime 0:37:48
numbas_lti:numbas_lti_workers_00   RUNNING   pid 2001, uptime 0:37:48
numbas_lti:numbas_lti_workers_01   RUNNING   pid 2002, uptime 0:37:48
```

Reporting scores back to the VLE fails

Check that the server running the LTI provider can make HTTPS requests to the address your VLE is accessed through.

If the connection always times out or there's an error making a connection, it may be the case that HTTPS connections are blocked.

If you get any other error, check that the Numbas LTI provider is reporting back to the right address. You can find the address used for a particular resource using the Django shell. First, find the ID of the *resource* you're interested in - it's the numerical part of the address when you're looking at a resource's dashboard.

In a terminal, navigate to the Numbas LTI provider's directory and run the following (commands after >>> are Python code):

```
python manage.py shell
>>> from numbas_lti.models import LTIUserData
>>> resource_pk = 1 # Replace this with the ID of the resource you're interested in.
>>> print(LTIUserData.objects.filter(resource__pk=resource_pk).first().lis_outcome_
↪service_url)
```

Check with your VLE's documentation that this is the right address.

1.4.6 Upgrading to a new version

When upgrading the Numbas LTI provider, follow the upgrade instructions for your platform.

Sometimes new versions of the LTI provider require changes that can't be made automatically by the normal upgrade procedure.

For such releases, this page lists the changes that must be made.

v3.0

Docker installation

There are a few new settings which must be present in `settings.env`. See *Settings* for information on the values these settings should take.

- `INSTANCE_NAME`
- `TIME_ZONE`
- `DEFAULT_FROM_EMAIL`
- `SUPPORT_NAME`
- `SUPPORT_URL`
- `EMAIL_COMPLETION_RECEIPTS`

Non-Docker installation

This version updates many of the packages that the LTI tool relies on, and so introduces quite a few changes to the way that the tool is configured.

The minimum required versions of some software have increased:

- Python 3.8 or newer
- Redis 5 or newer

Git branch

There are now stable git branches for each major version of the LTI provider. When upgrading to v3.0, switch to the `v3_STABLE` branch:

```
cd /srv/numbas-lti-provider
git fetch origin
git checkout v3_STABLE
```

After this, you can proceed with the rest of the update steps for your system. For Ubuntu, the next command will be `source /opt/numbas_lti_python/bin/activate`.

Packages to install

There are changes to the required versions of packages specified in `requirements.txt`. In addition, if you are using Redis as the Channels backend, you will need to install `channels_redis`:

```
pip install channels_redis==3.3.1
```

Supervisor configuration

Overwrite the configuration for `supervisord` with the new one given in the installation instructions for your system: *Ubuntu*, *RHEL 7*.

The numbas_lti_worker processes are no longer needed, and the command to launch the numbas_lti_daphne processes has changed.

Changes to settings

There are several changes to make in the file numbasltiprovider/settings.py.

The template file, numbasltiprovider/settings.py.dist, has been rearranged and lots of comments added to better explain what the settings do. You might like to use that file as a base, and insert your existing settings, rather than just changing your existing file.

- The way that Channels is configured has changed. To use Redis as the backend, replace the CHANNEL_LAYERS setting with the following:

  ```
  CHANNEL_LAYERS = {
      "default": {
          "BACKEND": "channels_redis.core.RedisChannelLayer",
          "CONFIG": {
              "hosts": [os.environ.get('REDIS_URL','redis://localhost:6379')],
          }
      },
  }
  ```

  For other backends, see the Channels documentation.

- Django now requires the DEFAULT_AUTO_FIELD setting to be set as follows:

  ```
  DEFAULT_AUTO_FIELD='django.db.models.AutoField'
  ```

- If you are using MySQL for your database, add the following underneath inside the 'default' entry in the DATABASES setting, in order to improve handling of Unicode characters:

  ```
  'OPTIONS': {
      'charset': 'utf8mb4',
      'use_unicode': True,
  },
  ```

  You might need to convert the tables within MySQL to use the utf8mb4 character set and utf8mb4_unicode_ci collation rules. See this post by Mathias Bynens for instructions on how to do that.

- The Huey task runner now prioritises tasks. Change the HUEY setting to the following:

  ```
  HUEY = {
      'huey_class': 'huey.PriorityRedisHuey',
  }
  ```

- Add 'numbas_lti.context_processors.global_settings' to the TEMPLATES['OPTIONS']['context_processors'] setting.

- There is a new setting INSTANCE_NAME, which should contain the name of the server, to display to users. If the server is run by the University of Somewhere, you might set:

  ```
  INSTANCE_NAME = 'University of Somewhere'
  ```

- There is a new setting REPORT_FILE_EXPIRY_DAYS, specifying the number of days that report files should remain available, before being deleted. The recommended length of time to keep reports is 30 days:

```
REPORT_FILE_EXPIRY_DAYS = 30
```

- Remove `'django_cookies_samesite.middleware.CookiesSameSite'` from `MIDDLEWARE`.

v2.13

There was an error in the base `settings.py` file when localisation was introduced.

If you are using the English translation, in `numbasltiprovider/settings.py`, change `LANGUAGE_CODE = 'en-us'` to `LANGUAGE_CODE = 'en'`.

v2.11

This release adds a dependency on the django-statici18n package to translate dynamically-generated text.

In `numbasltiprovider/settings.py`, add `'statici18n'` to `INSTALLED_APPS`.

The whole list should now be:

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'channels',
    'huey.contrib.djhuey',
    'statici18n',
    'numbas_lti',
    'bootstrapform',
    'bootstrap_datepicker_plus',
]
```

v2.10

This release uses the task runner huey to perform long-running tasks.

In `numbasltiprovider/settings.py`, add `'huey.contrib.djhuey` to `INSTALLED_APPS`.

The whole list should now be:

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'channels',
    'huey.contrib.djhuey',
    'numbas_lti',
    'bootstrapform',
    'bootstrap_datepicker_plus',
]
```

Add a huey process to the supervisord configuration (in `/etc/supervisor/conf.d/numbas_lti.conf` on Ubuntu):

```
[program:numbas_lti_huey]
command=/opt/numbas_lti_python/bin/python /srv/numbas-lti-provider/manage.py run_huey␣
↪-w 8
directory=/srv/numbas-lti-provider/
user=numbas_lti
autostart=true
autorestart=true
redirect_stderr=True
stopasgroup=true
environment=DJANGO_SETTINGS_MODULE="numbasltiprovider.settings"
numprocs=1
process_name=%(program_name)s_%(process_num)02d
stderr_logfile=/var/log/supervisor/numbas_lti_huey_stderr.log
stdout_logfile=/var/log/supervisor/numbas_lti_huey_stdout.log

[group:numbas_lti]
programs=numbas_lti_daphne,numbas_lti_workers,numbas_lti_huey
priority=999
```

Note that the `[group:numbas_lti]` section has changed as well.

1.5 For administrators

1.5.1 Consumers

A *tool consumer* is an application which sends users to the Numbas LTI provider, and receives score data in return. In most cases, this will be a Virtual Learning Environment (VLE), such as Blackboard, Canvas or Moodle.

Only admin users on the LTI provider can create and manage consumers.

Adding a consumer

The first step is to add a consumer. Click the *Add a new LTI consumer* button.

You will be asked to give a unique _key_ to identify the consumer, and optionally the URL of the consumer's homepage. The key can be any text, but should uniquely identify the consumer and its purpose.

Once you have created a consumer, you are shown its *key* and *shared secret*. These are the pieces of information that the consumer will use to establish a secure connection with the Numbas LTI provider.

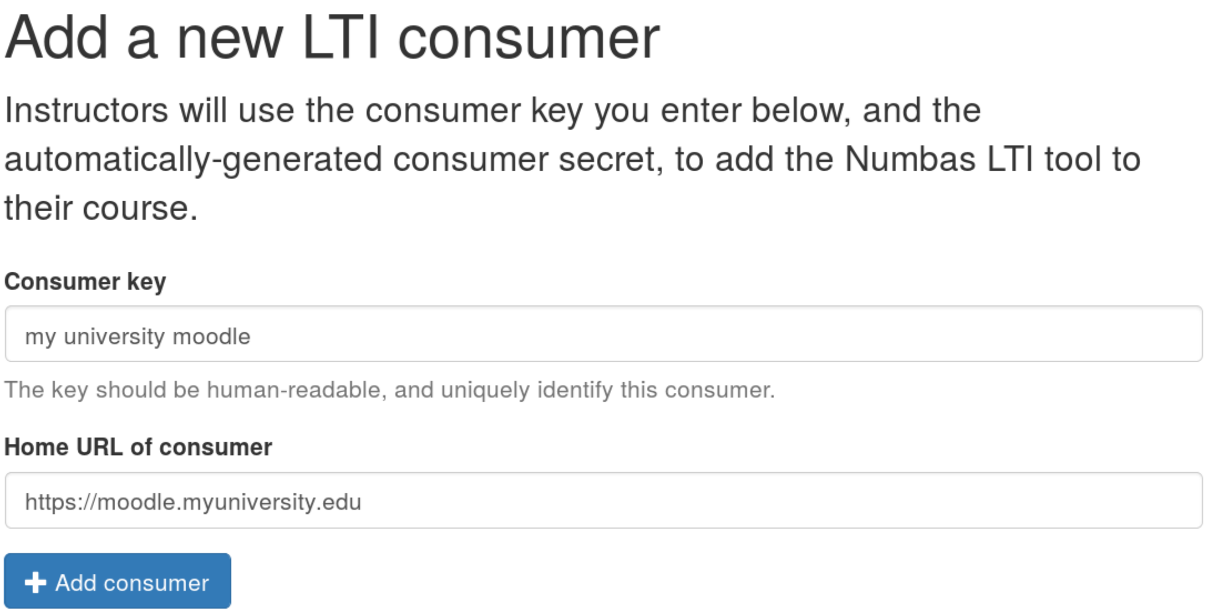
You must now set up the connection from the consumer's end. There are instructions for some of the most popular VLEs below.

Blackboard Learn

Follow Blackboard's instructions - you'll need to add Numbas as a new LTI Tool provider, and then add a placement for it.

Blackboard doesn't support automatic configuration by URL, so you'll have to use the manual configuration settings.

For the tool provider:



Add a new LTI consumer

Instructors will use the consumer key you enter below, and the automatically-generated consumer secret, to add the Numbas LTI tool to their course.

Consumer key

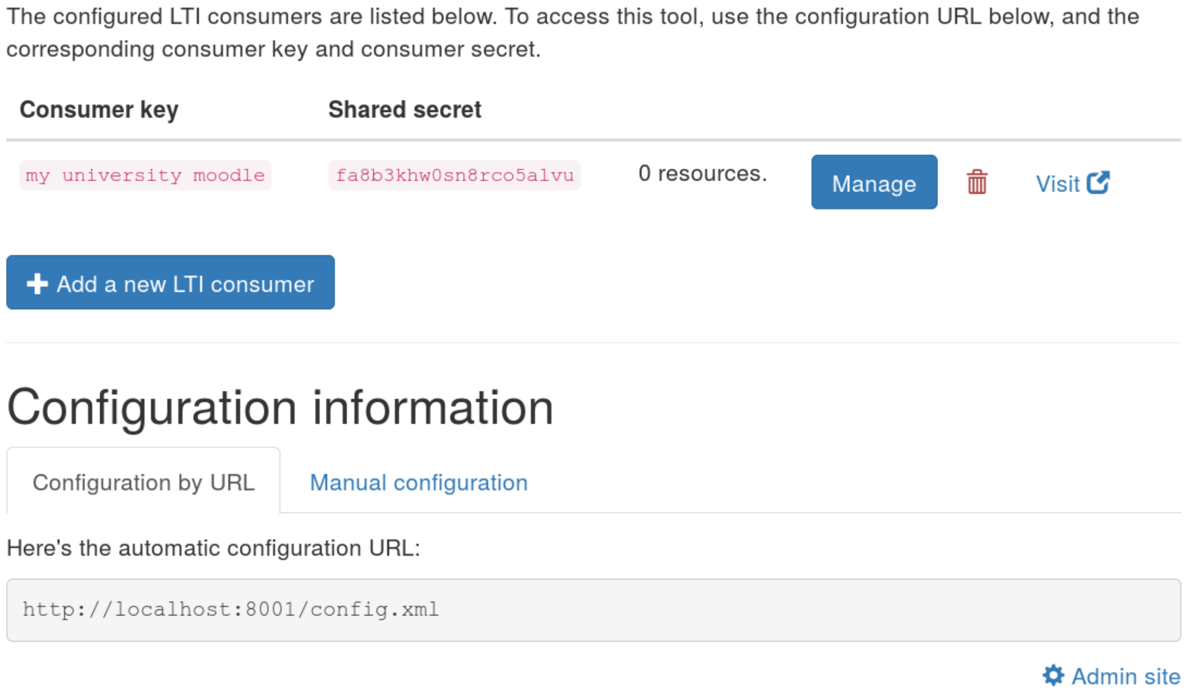
my university moodle

The key should be human-readable, and uniquely identify this consumer.

Home URL of consumer

https://moodle.myuniversity.edu

Fig. 3: The "Add a consumer" screen.



The configured LTI consumers are listed below. To access this tool, use the configuration URL below, and the corresponding consumer key and consumer secret.

Consumer key	Shared secret				
my university moodle	fa8b3khw0sn8rco5alvu	0 resources.	Manage	🗑	Visit

Configuration information

Here's the automatic configuration URL:

```
http://localhost:8001/config.xml
```

Fig. 4: The consumer management screen.

- The **Provider domain** is your launch URL.
- The **Tool provider key** is your consumer key.
- The **Tool provider secret** is your shared secret.

For the placement:

- The **Label** is "Numbas" (or whatever you want it to be)
- The **Type** is "Content Type", and select "Assessments" in the menu.
- Tick **Placement allows grading**.
- Save the icon from your "Manage consumers" page and upload it under the **Icon** field on the Blackboard form.

You can now add a Numbas exam to any folder in Blackboard by selecting "Numbas" from the *Assignments* menu.

Moodle

Moodle from version 3.0 onwards supports LTI configuration by URL. Earlier versions require you to enter the configuration settings manually.

If you have administrator privileges on Moodle, you can add the tool as an activity type, so you don't have to enter the settings for each exam.

Moodle 3.0+

- Go to *Site administration* → *Plugins* → *Activity modules* → *External tool* → *Manage tools*.
- Paste your **automatic configuration URL** into the box labelled *Tool URL*, and click *Add*.
- Enter your **consumer key** and **shared secret**, then click "*Save changes*".
- Go back to the *Manage tools* page and edit the Numbas tool.
- For *Default launch container*, select "New window".
- If you'd like Numbas to show up as an activity type, bypassing the "External tool" step, for *Tool configuration usage*, select "Show in activity chooser and as a preconfigured tool".

Now you can add a Numbas exam in any course by clicking *Add an activity or resource* and selecting *External tool*, then select "Numbas" from the *Preconfigured tool* dropdown.

Versions of Moodle before 3.0

- Go to *Site administration* → *Plugins* → *Activity modules* → *External tool* → *Manage tools*.
- Click on *Configure a tool manually*.
- Set *Tool name* to "Numbas".
- The *Tool base URL* is your launch URL.
- The *Consumer key* is your consumer key.
- The *Shared secret* is your shared secret.
- Under *Tool configuration usage*, select "Show in activity chooser and as a preconfigured tool".

Now you can add a Numbas exam in any course by clicking *Add an activity or resource* and selecting *Numbas*.

Canvas LMS

- Go to your course's *Settings* page, then click on *Apps* and *View App Configurations*.
- Click the button to add an App.
- For *Configuration type*, select "By URL".
- The *Consumer key* is your consumer key.
- The *Shared secret* is your shared secret.
- The *Config URL* is your automatic configuration URL.

Now you can add a Numbas exam in your *Modules* page:

- Click the button to add an item.
- Select *External tool* and click on *Numbas*.
- Click *Add item*.

Managing a consumer

Click on the *Manage* button to see more detailed information about a consumer.

The management view lists all contexts provided by this consumer. Contexts usually correspond to courses in a VLE - all *resources* launched from the same course will appear under the same context here. Click on a context's name to see all resources in that context.

Time periods

Time periods allow you to group a consumer's contexts by when they were active. You could use this to group contexts by academic year, for example.

Click on the *Manage time periods* button at the bottom of the consumer management page. Each time period has a name and start and end dates.

Each context is assigned to the latest time period containing the date on which the context's first resource was created. If a context falls into a gap between non-overlapping time periods, it will be shown separately.

Time periods are used solely to group contexts in the display. If you change or delete a time period, no data will be lost.

1.5.2 Editor links

Editor links form a connection between the Numbas LTI tool provider and an instance of the Numbas editor, allowing instructors to select exams from a list inside the LTI tool, rather than downloading an exam package to their own computer and then uploading it to the LTI tool.

The public Numbas editor hosted by mathcentre provides a wealth of ready-made exams on a variety of subjects.

Creating an editor link

From the *admin interface*, click on the *Editor links* button at the top of the screen.

Click on the *Connect to a Numbas editor* button. In the *Base URL of the editor* field, enter the address of the Numbas editor.

The address of the public editor hosted by mathcentre is:

```
https://numbas.mathcentre.ac.uk
```

Once you've created an editor link, you can select the projects from the editor that you want to use.

Only projects which are marked visible to non-members are available. This is because the editor link has no authentication mechanism, and linked projects are available to all *consumers*.

Once you have selected some projects, published exams from those projects will be available for instructors to pick on the *new Numbas activity* screen.

1.5.3 Stress tests

Use the stress test feature to see how the LTI provider copes with high demand. You can use this to identify potential problems before any sessions where you expect many students to be attempting a Numbas exam at the same time.

During a stress test, each device taking part will open multiple connections to the LTI provider and simulate sending attempt data.

Running a stress test

To begin a new stress test, click the *Start a stress test* button. You will be taken to the stress test control panel.

On each device taking part in the test, log in to the Numbas LTI provider's admin interface and navigate to the stress test's control panel.

You can simulate a single attempt by clicking the *Start an attempt* button. Each simulated attempt will open a connection from your device to the LTI provider, and send a new data element at the rate specified by the *Set new elements every __ seconds* field.

You can start multiple attempts at the same time with the first line of the form. Click the *now* button to start some attempts immediately, or enter a time - you could use this to coordinate several devices.

> **Caution:** Many web browsers have a limit on the number of websocket connections they can hold open at once.
>
> At time of writing, Chrome is limited to 255 simultaneous connections, Firefox to 200, and Internet Explorer is limited to 6.
>
> If you simulate more attempts than this, the extra attempts will use the AJAX fallback.

You can change the rate at which new elements are set, and the number of elements set when each attempt starts.

A Numbas exam normally sets around 100 elements when an attempt starts. This number grows slightly for exams with more questions and question parts.

When a student submits an answer, three or four elements are set. So, a "realistic" value for the *Set new elements every __ seconds*' field will be around $\frac{\text{total attempt time}}{5\times(\text{number of parts})}$.

Interpreting the results of a stress test

Summary is shown above the list of all simulated attempts. This is only for the attempts being simulated on your device - because the point of the test is to identify bottlenecks in the network, there's no reliable way of showing data about all devices taking part in the test. For each attempt, the current state of the connection, and the number of elements unsaved, are shown.

All attempts should have finished “waiting to start” within a minute or so of the test beginning.

Ideally, there should be zero attempts without a websocket connection. Attempts without a websocket connection will use the AJAX fallback, which saves data less regularly.

The most important statistic is the number of attempts with unsaved data. If this number remains above 0 for a long time, it means that the server has a backlog of data to save. In a real test, each device will keep resending data until it gets an acknowledgement that it has been saved, so data will eventually be recorded, if slowly.

If the connection is lost entirely - for example, if there’s a power cut, the student’s computer crashes, or the student closes the test before all data has been saved - any data which has not been acknowledged by the server will be saved to the browser’s local storage on the student’s machine. When the student reopens the test, any data in local storage will be sent.

If an attempt shows *AJAX is not working*, then a request to use the AJAX fallback has failed: usually because the connection timed out or was refused entirely. This is a sign that the LTI provider can not cope with the amount of traffic you’re simulating, and students would see errors or data might go missing in a real test.

1.5.4 Dashboard

The *Dashboard* shows all *resources* which have been launched in the past 24 hours. You can click on a resource’s name to view it, or go straight to its statistics page.

1.5.5 Search

The search box can be used to search for users, resources and contexts. It will search:

- Users’ full names
- Usernames (the identifier passed by the *tool consumer*)
- Users’ email addresses
- Resource names
- Exam package names (if different to the resource’s name(
- Context names
- Context labels

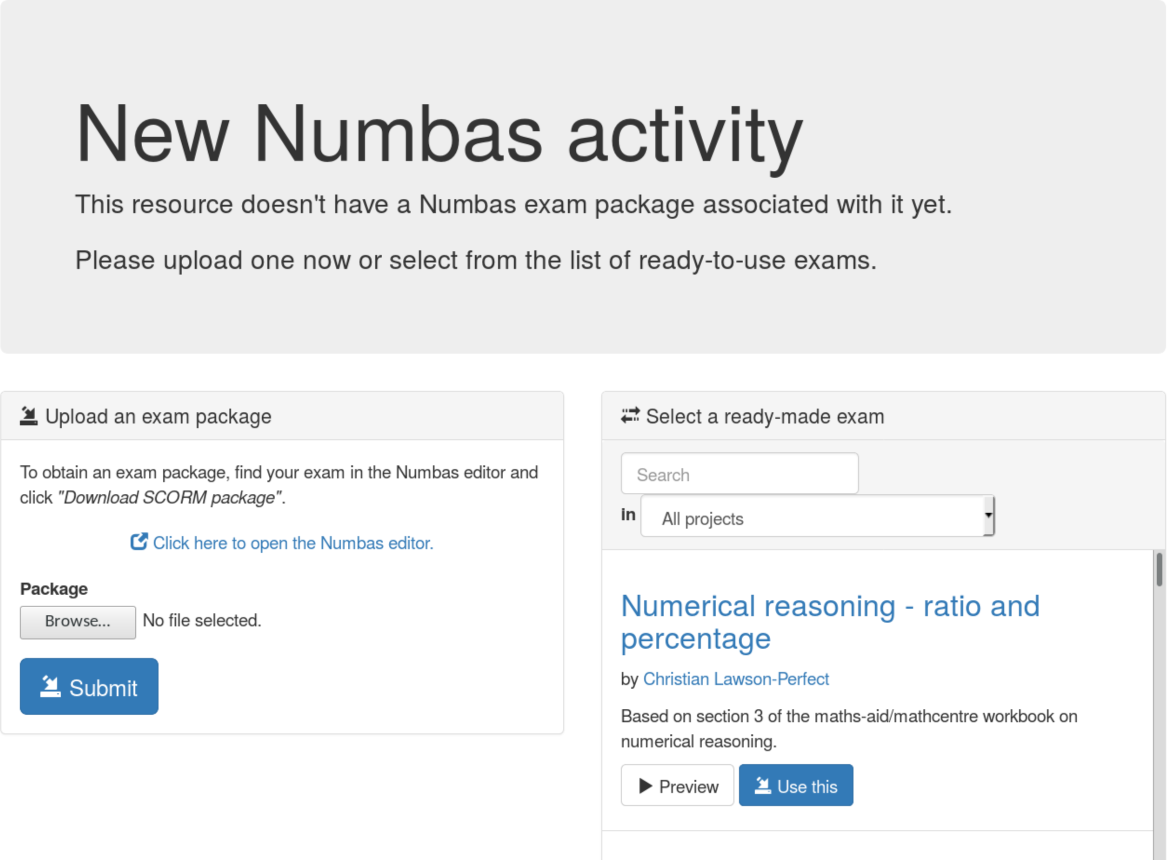
The search box shows some suggestions as you type; you can also press the *Submit* button to see a complete list of results.

1.6 For instructors

1.6.1 Resources

A *resource* is a single Numbas exam, which students access via the VLE.

Resources are automatically created when you launch a new Numbas activity from the VLE as an instructor. You must select a Numbas exam to use, and then any students who launch the same activity will be shown the exam.



New Numbas activity

This resource doesn't have a Numbas exam package associated with it yet.

Please upload one now or select from the list of ready-to-use exams.

Upload an exam package

To obtain an exam package, find your exam in the Numbas editor and click *"Download SCORM package"*.

Click here to open the Numbas editor.

Package

Browse... No file selected.

Submit

Select a ready-made exam

Search

in All projects

Numerical reasoning - ratio and percentage

by Christian Lawson-Perfect

Based on section 3 of the maths-aid/mathcentre workbook on numerical reasoning.

Preview Use this

Fig. 5: The first thing you see on creating a new Numbas resource.

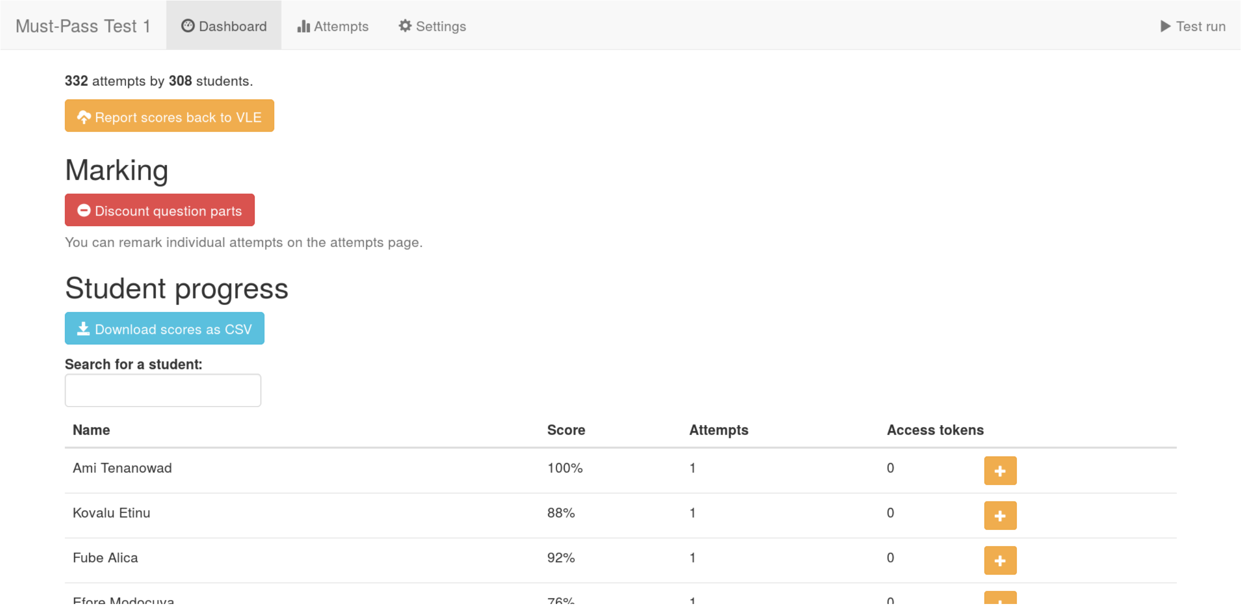
Creating a new resource

Either upload an exam package that you have downloaded from the Numbas editor or, if any *editor links* have been created, select an exam from the list.

When you download an exam package from the Numbas editor, you must use the *SCORM package* option.

Once you've selected an exam, you will be shown the dashboard for the resource.

Dashboard



332 attempts by **308** students.

You can remark individual attempts on the attempts page.

Name	Score	Attempts	Access tokens	
Ami Tenanowad	100%	1	0	+
Kovalu Etinu	88%	1	0	+
Fube Alica	92%	1	0	+
Efore Modocuva	76%	1	0	+

Fig. 6: The dashboard screen.

When you open a resource as an instructor, you are first shown the dashboard. This view offers a few actions to do with the resource, and the navigation bar at the top has links to other views.

Report scores back to VLE

The LTI provider can automatically report scores back to the VLE.

Click the *Report scores back to VLE* button to begin this process. This may take some time; you'll be shown either a success message or any errors encountered while reporting scores.

This isn't supported by every VLE.

Discount question parts

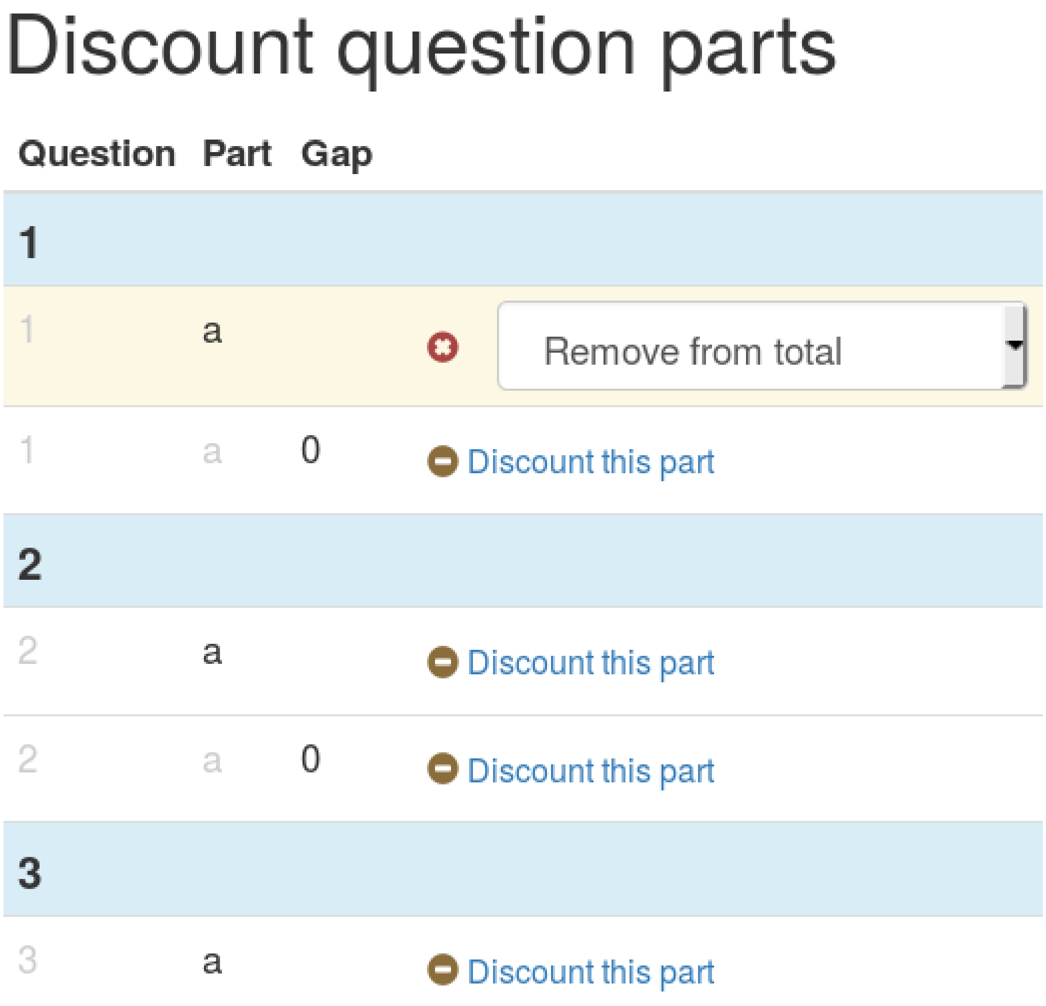
Discounting a question part removes it from the score calculations: any marks students have earned for that part are discounted.

You might want to do this if an error is found in a question.

To discount a part, click on the *Discount a question part* button on the dashboard.

You are shown a list of all the question parts in the exam. Click a *Discount this part* button next to a part to discount it. You can choose whether to remove the part from the total available for the exam, or to award everyone full marks for the part. These have different effects on the weighting of other parts in the exam - removing a part from the total will increase the weighting of other parts, while awarding full marks will increase everyone's total score. Think carefully about what you want to do.

If you discount a gapfill part, all of its gaps are discounted. If you discount an individual gap, the other gaps are unaffected.



Discount question parts

Question	Part	Gap	
1			
1	a		✖ Remove from total
1	a	0	⊖ Discount this part
2			
2	a		⊖ Discount this part
2	a	0	⊖ Discount this part
3			
3	a		⊖ Discount this part

Fig. 7: Discounting a question part.

Download scores as CSV

Click this button to download a `.csv` file containing the scores for each student who has attempt the resource.

The columns of the file are:

- First name
- Last name

- Email address
- Username
- Percentage score

The values in the name, email address and username fields come from the VLE. The username field might not correspond exactly to the student's username on the VLE; in particular, Blackboard preprends usernames with `cuid:`.

The file might take a while to produce; it'll be available in the *Reports* tab once it's ready.

Student progress

From the dashboard, click *View individual student progress and grant access tokens* to view the *Student progress* table.

The *Student progress* table lists the names of students who have attempted the activity, along with their scores as calculated following the *grading method*, and the number of attempts they have made.

You can narrow down the displayed list by entering a name in the *Search for a student* box.

Note that only students who have launched the activity are listed - the LTI provider has no way of knowing about students who have access to the activity through the VLE but have never launched it.

Access tokens

When the number of attempts students are allowed to make is limited, circumstances can arise in which you want to allow particular students another attempt.

To do this, click the plus symbol in the *Access tokens* column next to the student's name in the *Student progress* table.

To revoke an access token, click the minus symbol next to the student's name in the *Access tokens* column. Attempts started with an access token that is subsequently removed will not be deleted.

When the student launches the activity, they will be offered the opportunity to start a new attempt.

Validate a receipt code

If enabled (see *Email attempt receipts to students on completion?*), students are emailed a receipt on completion of an attempt. The receipt contains information about their attempt, such as time and score, as well as a code that you can use to confirm that a receipt is valid.

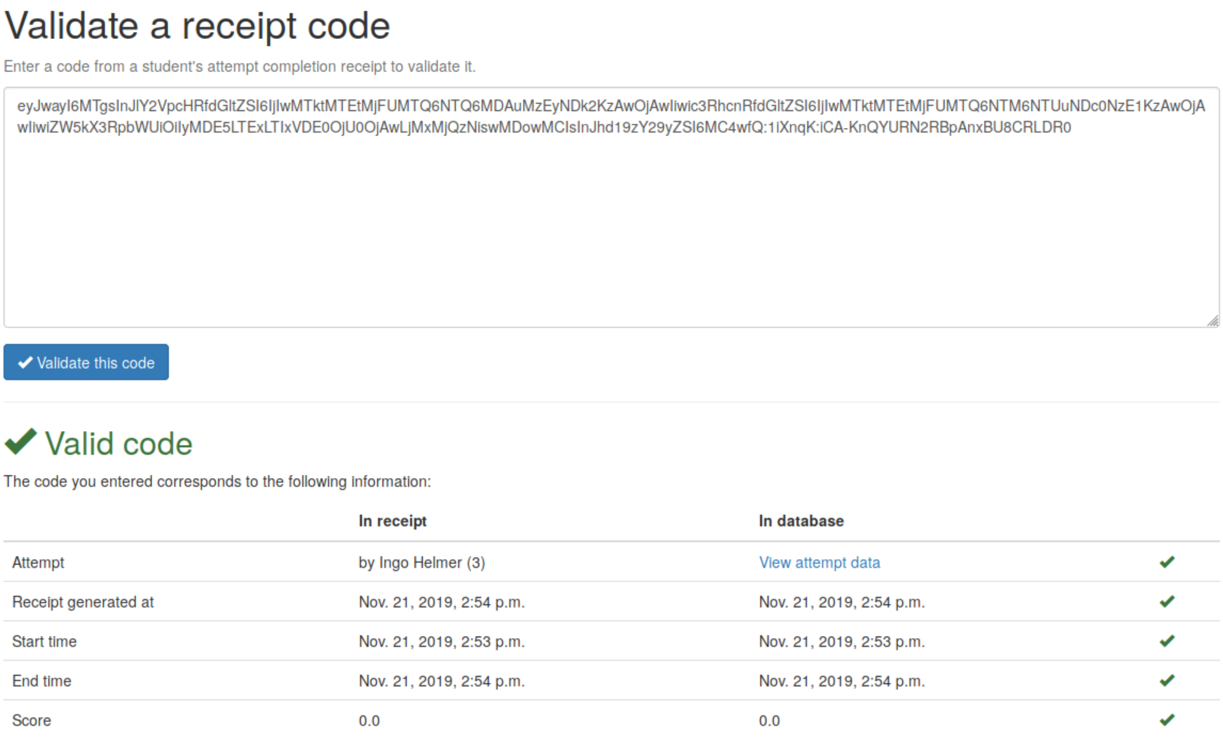
On the *Validate a receipt code* page, copy a code from a student into a box.

If the code is valid, you'll be shown the information from the receipt, as well as a link to view the attempt's data in more detail.

If the code is invalid, or corresponds to an attempt at a different resource, you'll be told so.

Attempts

Click on the *Attempts* button at the top of the page to view the attempt management screen.



Validate a receipt code

Enter a code from a student's attempt completion receipt to validate it.

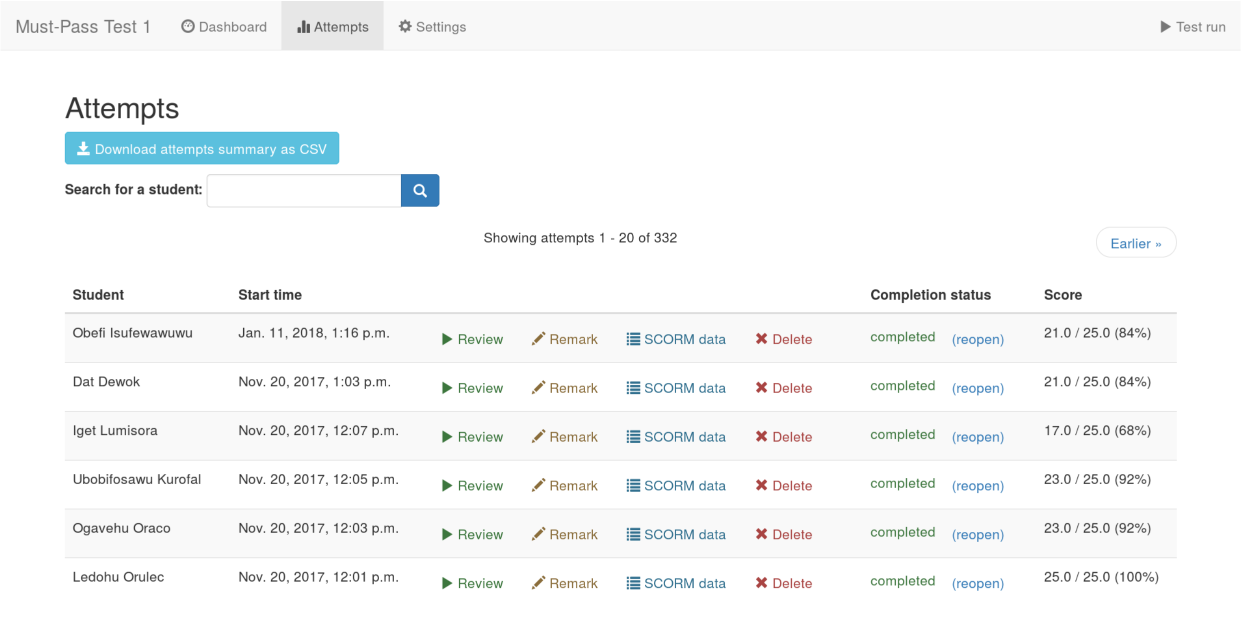
```
eyJwayI6MTgsInJlY2VpcHRfdGltZSI6IjIwMTktMTEtMjFUMTQ6NTQ6MDAuMzEyNDk2KzAwOjAwIiwic3RhcnRfdGltZSI6IjIwMTktMTEtMjFUMTQ6NTM6NTUuNDc0NzE1KzAwOjA
wIiwiZW5kX3RpbWUiOiIyMDE5LTExLTIxVDE0OjU0OjAwLjMxMjQzNiswMDowMCIsInJhd19zY29yZSI6MC4wfQ:1iXnqK:iCA-KnQYURN2RBpAnxBU8CRLDR0
```

✔ Valid code

The code you entered corresponds to the following information:

	In receipt	In database	
Attempt	by Ingo Helmer (3)	View attempt data	✔
Receipt generated at	Nov. 21, 2019, 2:54 p.m.	Nov. 21, 2019, 2:54 p.m.	✔
Start time	Nov. 21, 2019, 2:53 p.m.	Nov. 21, 2019, 2:53 p.m.	✔
End time	Nov. 21, 2019, 2:54 p.m.	Nov. 21, 2019, 2:54 p.m.	✔
Score	0.0	0.0	✔

Fig. 8: A validated receipt code.

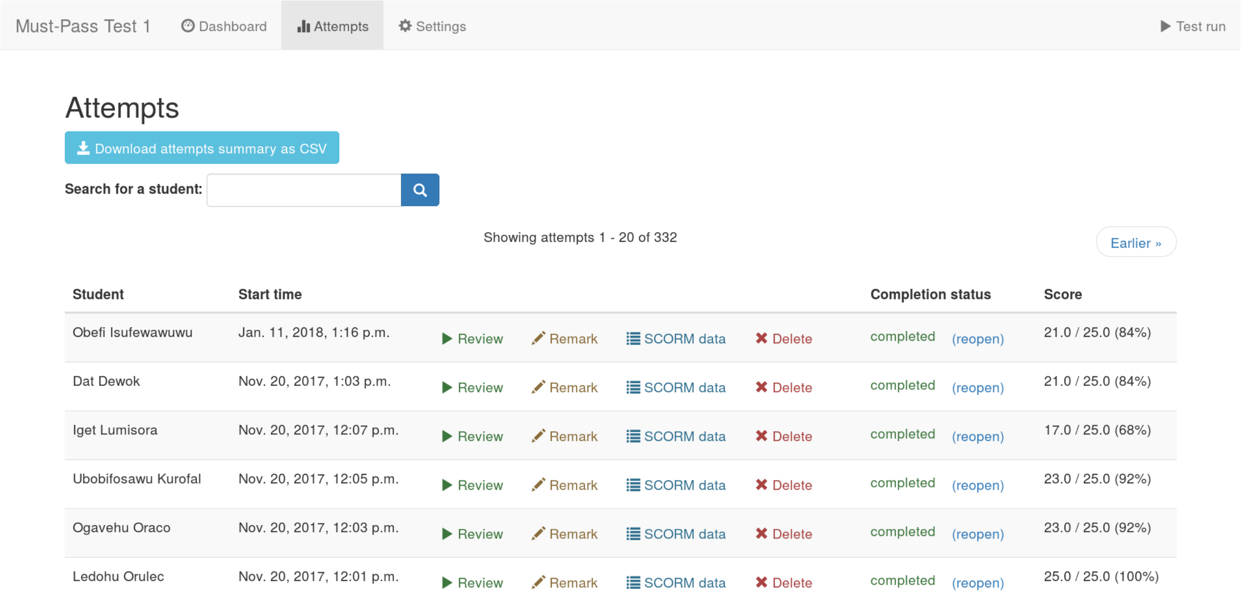


Attempts

Showing attempts 1 - 20 of 332

Student	Start time					Completion status		Score
Obefi Isufewawuwu	Jan. 11, 2018, 1:16 p.m.	▶ Review	Remark	SCORM data	✖ Delete	completed	(reopen)	21.0 / 25.0 (84%)
Dat Dewok	Nov. 20, 2017, 1:03 p.m.	▶ Review	Remark	SCORM data	✖ Delete	completed	(reopen)	21.0 / 25.0 (84%)
Iget Lumisora	Nov. 20, 2017, 12:07 p.m.	▶ Review	Remark	SCORM data	✖ Delete	completed	(reopen)	17.0 / 25.0 (68%)
Ubobifosawu Kurofal	Nov. 20, 2017, 12:05 p.m.	▶ Review	Remark	SCORM data	✖ Delete	completed	(reopen)	23.0 / 25.0 (92%)
Ogavehu Oraco	Nov. 20, 2017, 12:03 p.m.	▶ Review	Remark	SCORM data	✖ Delete	completed	(reopen)	23.0 / 25.0 (92%)
Ledohu Orulec	Nov. 20, 2017, 12:01 p.m.	▶ Review	Remark	SCORM data	✖ Delete	completed	(reopen)	25.0 / 25.0 (100%)

Fig. 9: An invalid receipt code.



Attempts

Search for a student:

Showing attempts 1 - 20 of 332

Student	Start time					Completion status		Score
Obefi Isufewawuwu	Jan. 11, 2018, 1:16 p.m.	▶ Review	Remark	SCORM data	✖ Delete	completed	(reopen)	21.0 / 25.0 (84%)
Dat Dewok	Nov. 20, 2017, 1:03 p.m.	▶ Review	Remark	SCORM data	✖ Delete	completed	(reopen)	21.0 / 25.0 (84%)
Iget Lumisora	Nov. 20, 2017, 12:07 p.m.	▶ Review	Remark	SCORM data	✖ Delete	completed	(reopen)	17.0 / 25.0 (68%)
Ubobifosawu Kurofal	Nov. 20, 2017, 12:05 p.m.	▶ Review	Remark	SCORM data	✖ Delete	completed	(reopen)	23.0 / 25.0 (92%)
Ogavehu Oraco	Nov. 20, 2017, 12:03 p.m.	▶ Review	Remark	SCORM data	✖ Delete	completed	(reopen)	23.0 / 25.0 (92%)
Ledohu Orulec	Nov. 20, 2017, 12:01 p.m.	▶ Review	Remark	SCORM data	✖ Delete	completed	(reopen)	25.0 / 25.0 (100%)

Fig. 10: The attempt management screen.

Download attempts summary as CSV

Click the *Download attempts summary as CSV* button to obtain a `.csv` file with information on every attempt at this activity.

The columns of the file are:

- First name
- Last name
- Email address
- Username
- Start time, in `YYYY-MM-DD HH:MM:SS.ffffff+HH:MM` format.
- Completed? (Either `completed` or `incomplete`)
- Total score
- Percentage (total score as a percentage of marks available)
- One column giving the total score for each question

The file might take a while to produce; it'll be available in the *Reports* tab once it's ready.

Download all attempt data as JSON

Click the *Download all attempt data as JSON* button to obtain a `.json` file containing all data to do with attempts on this resource, including part and question scores, all SCORM data, completion status, and the values of question variables.

The file might take a while to produce; it'll be available in the *Reports* tab once it's ready.

Review an attempt

Click the *Review* button to view a student's attempt as they saw it.

This is useful when a student queries the mark they were awarded for a part of the exam.

Note that review mode always opens attempts as if they were completed, even if the student has not yet ended the exam.

Remark an attempt

Click the *Remark* button to manually change the score awarded for a question part.

You are shown a list of every question part in the exam. Click the pencil icon on the row corresponding to the part you want to change, and enter the new score. The new score is saved as you type, and the totals for the question and the whole exam are recalculated automatically.

Note: After updating the exam package associated with a resource, you can *automatically remark all attempts* using the *Remark* tab.

Data

By clicking on the *Data* button, you can see a timeline describing actions the student took during their attempt.

Items shown include moving between questions, submitting answers, and marks awarded. This page will update in real-time as the student continues their attempt.

Click the *Download attempt data in JSON format* button to obtain a `.json` file containing all data pertaining to this attempt, including question and part scores, completion status, and the values of question variables.

Numbas uses the SCORM standard to store data about attempts. By clicking on the *View raw SCORM data for this attempt* button, you can see all of the SCORM data model elements stored for a particular attempt.

This is most useful for debugging connection errors, to confirm that data has been saved.

If *Most recent value only* is ticked, only the most recent value for each element is shown. Untick it to see every value that the element has taken since the start of the attempt.

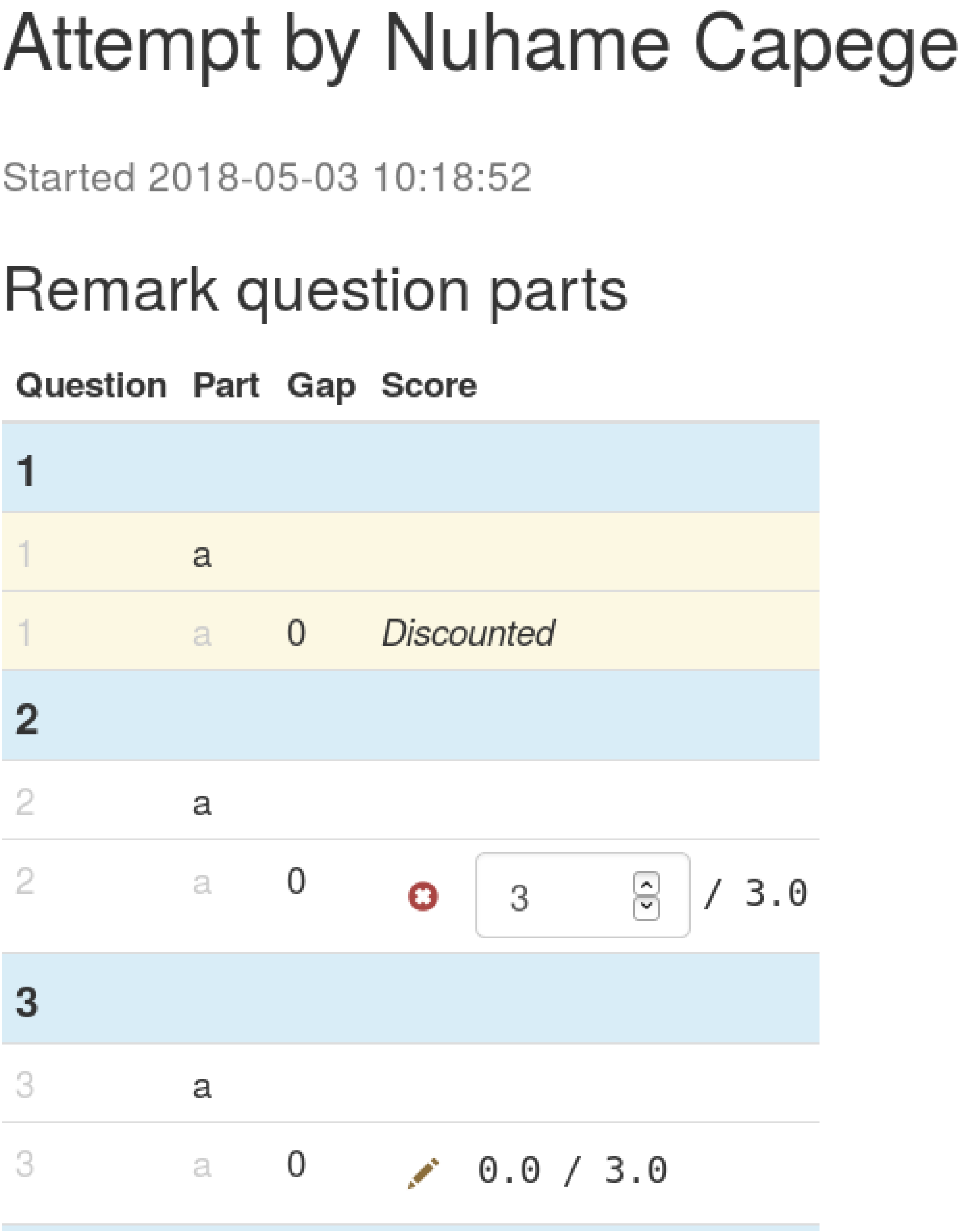
You can type a regular expression in the *Search for an element* box to narrow down the displayed list of elements.

Delete an attempt

Click the *Delete* button to delete an attempt. This is permanent; the student will be able to start a new attempt next time they launch the activity.

Reopen an attempt

Sometimes students accidentally close their attempts before they mean to. Click the *Reopen* button to allow a student to complete their attempt. The next time that they launch the activity, they will be able to resume the attempt as if they had only paused it.



Attempt by Nuhame Capege

Started 2018-05-03 10:18:52

Remark question parts

Question	Part	Gap	Score
1			
1	a		
1	a	0	*Discounted*
2			
2	a		
2	a	0	✖ 3 / 3.0
3			
3	a		
3	a	0	✎ 0.0 / 3.0

Fig. 11: Remarking an attempt.

Beware that the standard Numbas settings allow a student to see the correct answers to every question once they have finished their attempt. If you're concerned about this, it's often better to make the student start a new attempt, rather than reopen the previous one.

Statistics

The *Statistics* page shows some statistics derived from attempts at this resource.

Attempt completion

The *Attempt completion* table shows the completion status of attempts. Don't worry if some attempts are still marked as "incomplete" after the deadline has passed - scores for incomplete attempts are still counted.

An attempt will have the status "Not attempted" if the student opened the resource, but did not click the "Start exam" button. This can happen if the student's device has a problem which causes the exam not to load, or if it does load but they just don't start!

Summary statistics

The *Summary statistics* table shows the mean, median and quartiles for some statistics about the resource. The *Time taken* row is a rough measure of how long the student spent in their attempt; see *Time spent* for more information. Be careful when using this measurement: if a student completes their attempt in several sittings, the whole intervening time will be included, and even when a student has an attempt open, they might not be giving it their attention for the whole time.

Next is a breakdown of scores at each question, as a bar chart. The attempts at each question are classified as "not attempted", "incorrect" (score 0), "partially correct", or "correct" (the maximum score for the question).

Score distribution

The distribution of scores for the whole exam and for each question are shown as decreasing plots. The horizontal axis represents percentage score, and the vertical axis represents the proportion of attempts achieving at least that score.

Attempt times

Each attempt at the resource is shown as a span of time, with a dot at the recorded start time and another at the recorded end time.

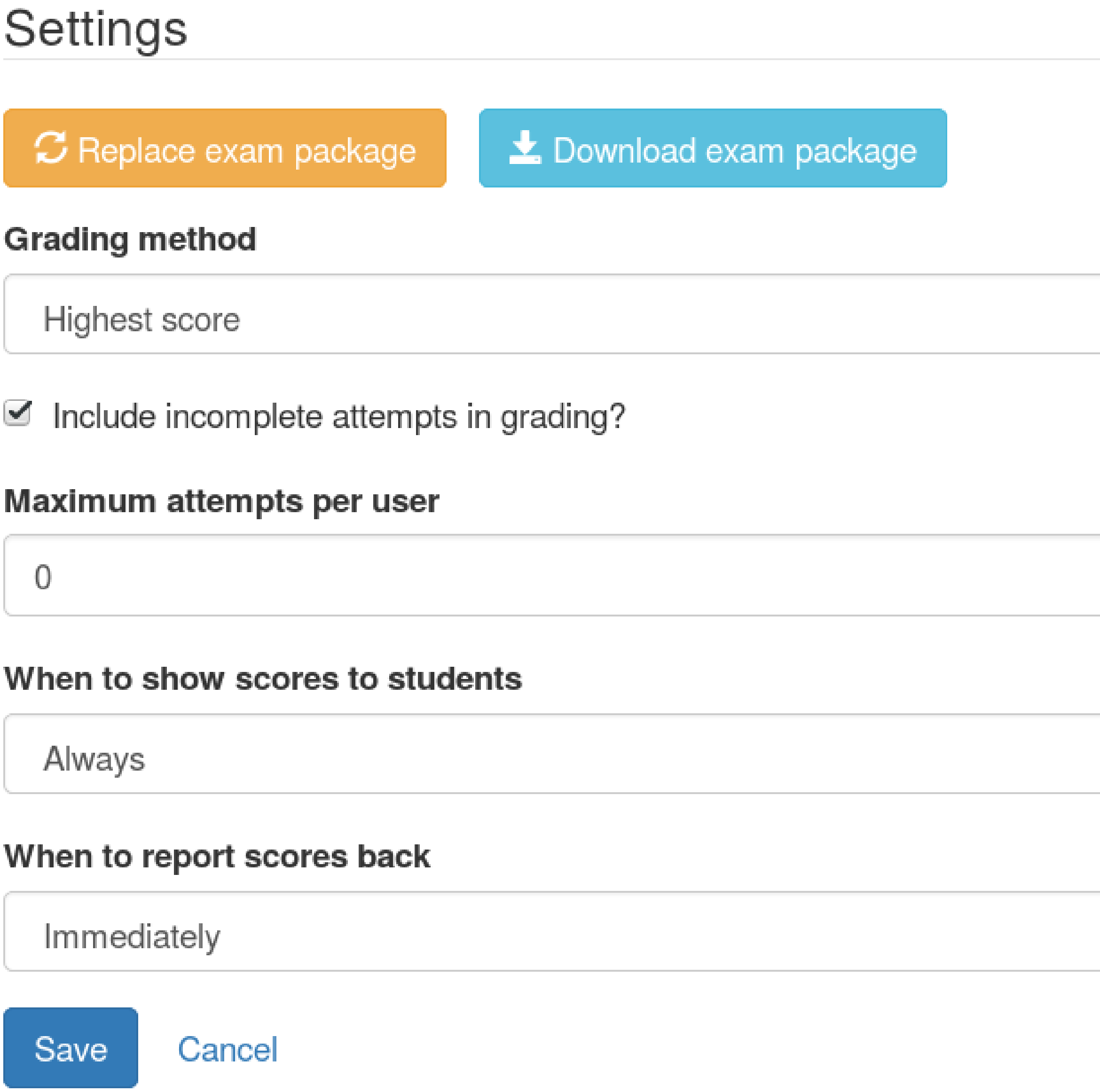
Time spent

The distribution of time spent at attempts is shown as a decreasing plot. The horizontal axis represents lengths of time, and the vertical axis represents the proportion of attempts at least that long.

The 'time spent' is a rough measure of how long the student spent in their attempt. This time is increased each time attempt data is saved, based on the time since their current session started. It is not simply the difference between the start and end times: if the student closes their attempt for a long time and then resumes it later, the time that it was closed is not counted.

Be careful when interpreting this measurement: students might leave an attempt open while they do something else, so there are often large outliers.

Settings



Settings

Grading method

Highest score

☑ Include incomplete attempts in grading?

Maximum attempts per user

0

When to show scores to students

Always

When to report scores back

Immediately

Fig. 12: The resource settings screen.

Replace exam package

If you discover an error in your exam, you can update it by downloading it again from the editor and clicking the *Replace exam package* button.

Any new attempts will use the latest version of the exam package. Because the new version might have changed in a way that is incompatible with existing attempts, for example by removing or rearranging question parts, any attempts started with the old package will by default continue to use the old package. If you know that the new package is compatible with the old one, for example if you've just corrected some text or fixed a bug in some code rather than changing the structure of the exam, tick *Make existing attempts use this version*. All attempts using the old package will be updated to use the new one.

If the new version of the exam fixes problems with marking, you can try *automatically remarking existing attempts*.

Grading method

Specify how a student's score for the activity is calculated.

- "Highest score" will use the highest total score from any of the student's attempts.
- "Last attempt" will use the total score from the attempt which the student began last.

Include incomplete attempts in grading?

If ticked, incomplete attempts will be included when calculating the student's score for the activity.

It's normally good to leave this on, so that students who forget to click the *End Exam* button won't be penalised.

Maximum attempts per user

How many *attempts* at the resource can each user take?

If set to 0, then there is no limit.

You can grant extra attempts to individual students with *access tokens*.

When to show scores to students

When a student reopens an activity, they are shown a summary of their attempts. You might not want to immediately show students their scores on this screen.

- "Always" means the student will see scores for all attempts, including incomplete attempts.
- "When attempt is complete" means the student will only see their score for an attempt once it is complete.
- "When review is allowed" means the student will only see their score after the date specified in the *Allow students to review attempts from* setting.
- "Never" means that no scores are shown to the student, even after they've completed their attempt.

Warning: This only controls the display of scores by the LTI provider. If you want to hide scores from the students, you must also turn off the score feedback options in the exam editor.

When to report scores back

Specify when students' scores are reported back to the *consumer*. Some VLEs make reported scores available to students immediately, which you may not want.

- "Immediately" - scores are reported as soon as they change, i.e. whenever a student submits an answer.
- "On completion" - a student's score is reported when they complete an attempt.

- “Manually, by instructor” - Scores are only reported when an instructor clicks the *Report scores back to VLE* button on the dashboard.

Allow students to review attempts from

Specify when students are allowed to re-enter completed attempts in review mode.

If left blank, students can review their attempts at any time.

If a date and time are set, students may only review their attempts after that time.

Instructors may always review students’ attempts, from the *Attempts* tab.

Available from and available until

Specify when students are allowed to access the resource.

You can set none, one or both of the options *Available from* and *Available until*.

If *Available from* is before *Available until*, the resource will be available only between those two times. Use this for a resource which should only be available for a certain period, such as an exam.

If *Available from* is after *Available until*, the resource will be *unavailable* between those two times, but available any time before *Available until* and any time after *Available from*. Use this for a resource which should be made unavailable for a certain period, such as a bank of practice material which students should not have access to during a summative assessment period.

Any students who have the resource open when it becomes unavailable will be forced to quit.

Email attempt receipts to students on completion?

If ticked, then when a student completes an attempt at this resource they will be emailed a receipt summarising their attempt.

The receipt contains a code which instructors can use to confirm the receipt’s contents. See *Validate a receipt code*.

Remark

The *Remark* tab provides an interface for automatically re-running attempts at the resource using the latest version of the exam package. If the total score awarded for an attempt is different to the saved score, you can overwrite it.

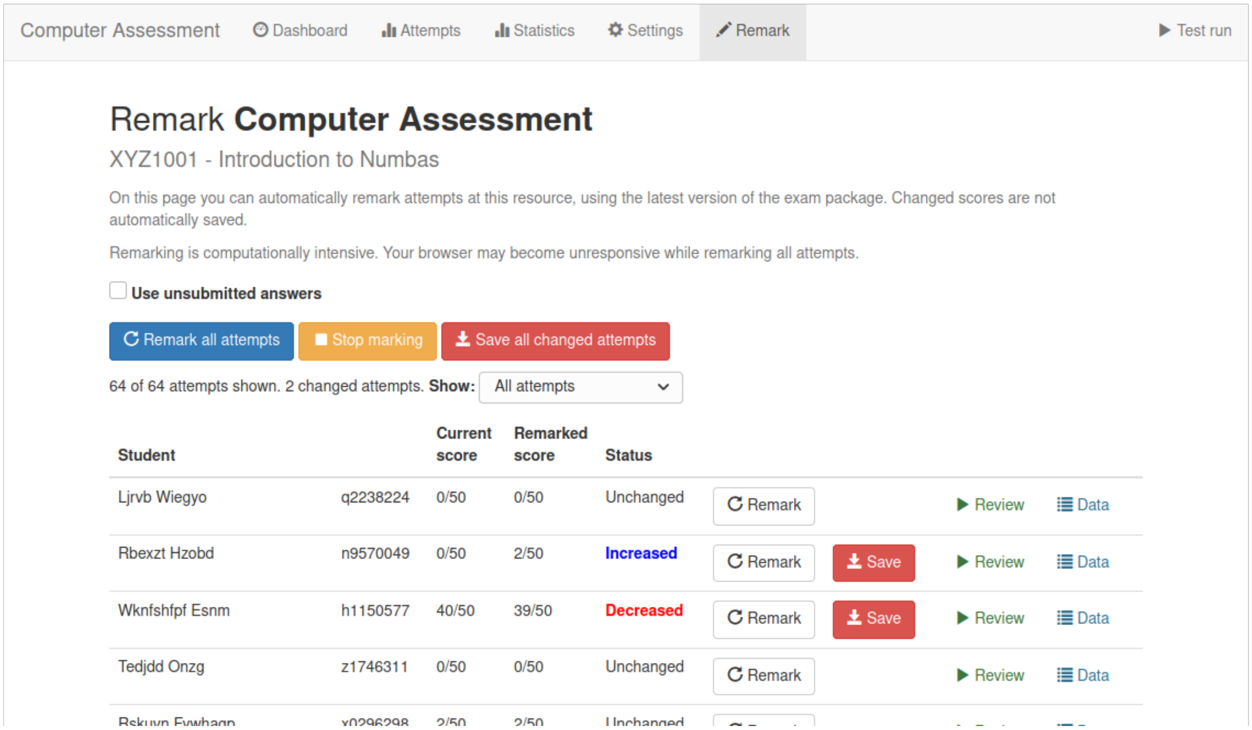
When the page loads, you are shown all attempts at the resource. You can remark individual attempts, or remark all the attempts automatically.

Note: Remarking an exam is computationally intensive, so your browser may appear unresponsive.

If *Use unsubmitted answers* is ticked, then any answers entered by the student but not submitted will be considered to be submitted. You can use this in cases where a student forgot to submit their answers and you want to have them marked anyway.

After you click *Remark all attempts*, each attempt will be remarked in turn. A progress bar shows the proportion of attempts that have been remarked, and an estimated time until completion.

To stop the process of remarking all attempts, click *Stop marking*. If an attempt is currently being remarked, it can’t be interrupted - the process will stop after that attempt is finished.



Remark **Computer Assessment**

XYZ1001 - Introduction to Numbas

On this page you can automatically remark attempts at this resource, using the latest version of the exam package. Changed scores are not automatically saved.

Remarking is computationally intensive. Your browser may become unresponsive while remarking all attempts.

☐ **Use unsubmitted answers**

64 of 64 attempts shown. 2 changed attempts. **Show:** All attempts

Student		**Current score**	**Remarked score**	**Status**				
Ljrvb Wiegyo	q2238224	0/50	0/50	Unchanged	Remark		Review	Data
Rbexzt Hzobd	n9570049	0/50	2/50	**Increased**	Remark	Save	Review	Data
Wknfshfpf Esnm	h1150577	40/50	39/50	**Decreased**	Remark	Save	Review	Data
Tedjdd Onzg	z1746311	0/50	0/50	Unchanged	Remark		Review	Data
Rskuyn Fywhagn	x0296298	2/50	2/50	Unchanged				

Fig. 13: Remarking a resource.

You can choose to *Show* all attempts, only attempts with changed scores, or only attempts with either increased or decreased scores.

When one or more attempts have been remarked and produced different scores, the *Save all changed attempts* button becomes available. After clicking this button, any changed data corresponding to the shown attempts is saved to the database. This data will appear in the *timeline for the attempt* and will affect the reported scores for the attempt.

Each attempt has its own *Remark* button, which will remark only that attempt. When an attempt has been remarked, the change in total score is shown. If the total score is different to that stored in the database, a *Save* button appears. Clicking the button causes changed data corresponding to the attempt to be saved to the database, updating its timeline and reported score.

Access changes

Access changes are a means of changing deadlines or allowing a different number of attempts at the resource to certain students.

Each access change applies to a list of students.

To create an access change, click on the *Access changes* tab, then *Add an access change*.

There are several fields that you can modify. Leave a field entirely blank to keep the standard value from the resource's settings.

In the *Description* field, describe what the access change is for and who it applies to, such as “25% extra time”, or “Late submission for Elliot D”.

Access changes take effect immediately. Any students who are completing an attempt at the time that their access changes will receive the new details immediately. If the resource becomes unavailable for them, the attempt will end immediately. Otherwise, a message will appear on their screen describing the new deadline.

Availability dates

The fields in the *Availability dates* section change the time period in which the affected students can complete attempts at the resource. You can either set new fixed start and end dates, or fill in the *Extend the deadline by* field to extend the resource's normal *Available from* date.

If several access changes apply to a student, the changes from the last created access change are used.

Exam duration

If the exam associated with the resource has a duration set, then you can extend it, either by an absolute length of time or by a percentage of the standard duration.

If several access changes apply to a student, the change in the last created access change is used.

Number of attempts

The *Maximum attempts per user* field overrides the resource's *Maximum attempts per user* field.

The student is allowed whichever is the greatest of the resource's normal maximum number of attempts and the numbers specified by any access changes affecting the student.

A value of zero in this field does not mean "no change", it means that the affected students are allowed to start as many attempts as they like.

Applies to

Specify which students the access change applies to by giving lists of usernames or email addresses. These are compared with the usernames and email addresses sent by the LTI consumer.

> **Warning:** Note that the LTI consumer can choose what it send for these fields, so they might not match the values you use in other systems. The access change form shows the username and email address it received from the LTI consumer for your account, as a guide.

Reports

When you ask to download one of the resource-level reports, it's compiled in the background. Once it's ready, you can download it from the *Reports* tab.

Reports are automatically deleted after a fixed period of time set by the administrator; the default is 30 days.

Test run

Click the *Test run* button to launch the Numbas exam. Data will not be saved - this feature is solely a convenience for instructors to check the contents of the exam.

1.7 For students

When a student clicks on a link to launch a Numbas *activity* for the first time, they are taken straight into the exam.

Their answers and scores are saved as soon as they are submitted.

> **Warning:** Unsubmitted answers are saved but not marked, so they will not receive any marks for answers not submitted before a resource's deadline. The automatic remarking tool can mark unsubmitted answers, if necessary. It's important to make this clear to students during high-stakes tests, because they may want to wait until they've finished the whole question or exam before submitting. Numbas allows students to submit answers as many times as they like, with no penalty, so encourage them to submit as soon as they've entered an answer.

If the student leaves the test before completing it, they will be able to resume from where they left off next time they return.

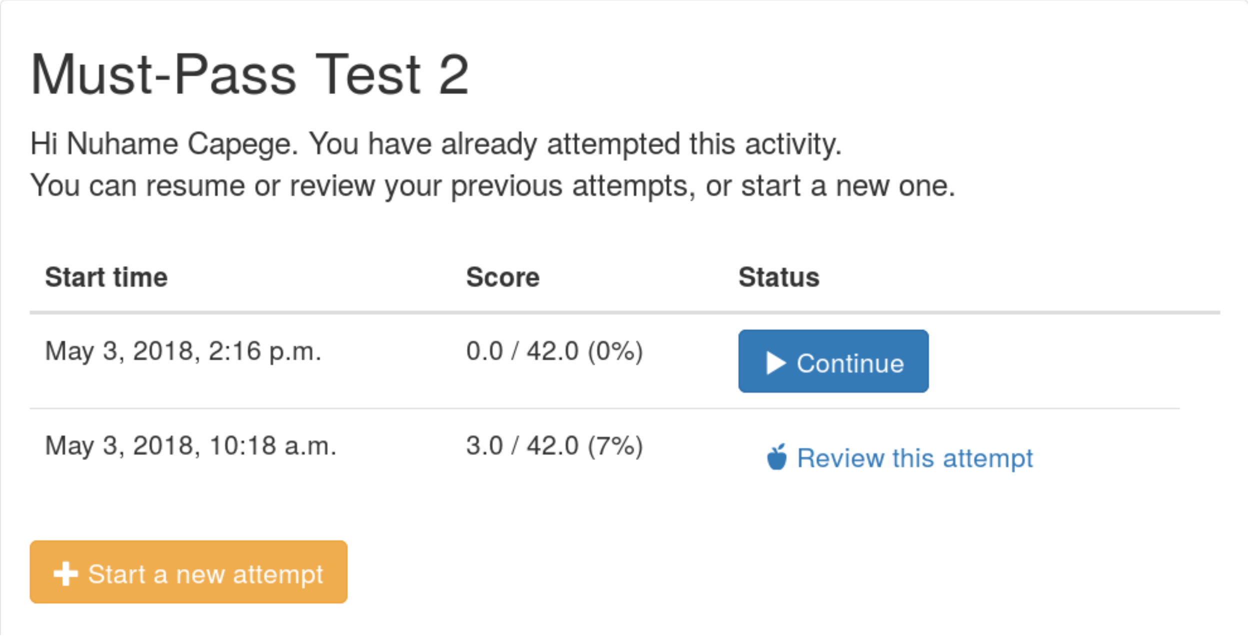
On subsequent launches of the activity, the student is shown a summary of their attempts. If *the exam's settings* allow it, the score earned for each attempt is shown.

The student can click the *Continue* button to resume an incomplete attempt where they left off.

Once a student clicks the *End exam* button, the attempt is marked as "completed", and the student can not change their answers. If the server is configured to do it, the student is emailed a receipt containing information about their attempt, and a code that can be used by the instructor to *validate it*.

For completed attempts, the student can click *Review this attempt* to reopen the attempt in review mode, the same as was shown when they ended the exam. They will see the results summary screen, and can click on individual questions to see their answers and any feedback.

If *the exam's settings* allow it, or the student has an unused *access token*, they can start a new attempt by clicking the *Start a new attempt* button.



Must-Pass Test 2

Hi Nuhame Capege. You have already attempted this activity.
You can resume or review your previous attempts, or start a new one.

Start time	Score	Status
May 3, 2018, 2:16 p.m.	0.0 / 42.0 (0%)	▶ Continue
May 3, 2018, 10:18 a.m.	3.0 / 42.0 (7%)	Review this attempt

Fig. 14: The attempt summary screen shown to a student.

1.8 Glossary

LTI Learning Tools Interoperability is a standard which defines how a *Tool consumer* connects to a *tool providing* a learning activity. The consumer can provide some information about the person who launched the activity, such as their name, their role in the course, and a unique identifier. The *Tool provider* can report back a score after a user has completed the activity.

VLE A Virtual Learning Environment, such as Blackboard, Moodle or Canvas. Also known as a Learning Management System, or LMS.

Tool consumer A system from which users launch LTI activities, such as a *VLE*. See *Consumers*.

Tool provider A system which provides activites to *tool consumers*. The Numbas LTI software is a tool provider.

Resource In the Numbas LTI provider, each link from a tool consumer to the provider creates a new resource, representing a single exam to be completed by students. See *Resources*.

Attempt A student may make one or more attempts at a *Resource*. Each attempt is independent from the others.

Editor link A connection between the LTI provider and an instance of the Numbas editor. See *Editor links*.

Index