MindMath LIP6 Server Manual

Editor: Yan Wang

Last Modified Date: 14/10/2020

Contents

[MindMath LIP6 Server Manual 1](#_Toc53583026)

[A. Continuous development of MindMath LIP6 server 3](#_Toc53583029)

[1. Install Java 3](#_Toc53583030)

[2. Install Git 4](#_Toc53583031)

[3. Install Eclipse 4](#_Toc53583032)

[4. Clone the project 5](#_Toc53583033)

[5. Switch the branch 6](#_Toc53583034)

[6. Import Gradle project with Eclipse 6](#_Toc53583035)

[7. Install dependencies via Gradle 7](#_Toc53583036)

[8. Run the server at localhost 8](#_Toc53583037)

[9. Generate WAR file and deployment 9](#_Toc53583038)

[10. Unit Test 9](#_Toc53583039)

[11. Generate SQL schema 11](#_Toc53583040)

[12. Update the images in MindMath server 11](#_Toc53583041)

[B. MindMath LIP6 APIs 13](#_Toc53583042)

[1. Task Controller 13](#_Toc53583043)

[2. Learning Controller 14](#_Toc53583044)

[3. File Controller 15](#_Toc53583045)

[4. Feedback Controller 15](#_Toc53583046)

[5. Other APIs: 16](#_Toc53583047)

[C. Documentation for LRS 18](#_Toc53583048)

[Statement 1 18](#_Toc53583049)

[Statement 2 19](#_Toc53583050)

[Statement 3 20](#_Toc53583051)

## ****Continuous development of MindMath LIP6 server****

### ****Install Java****

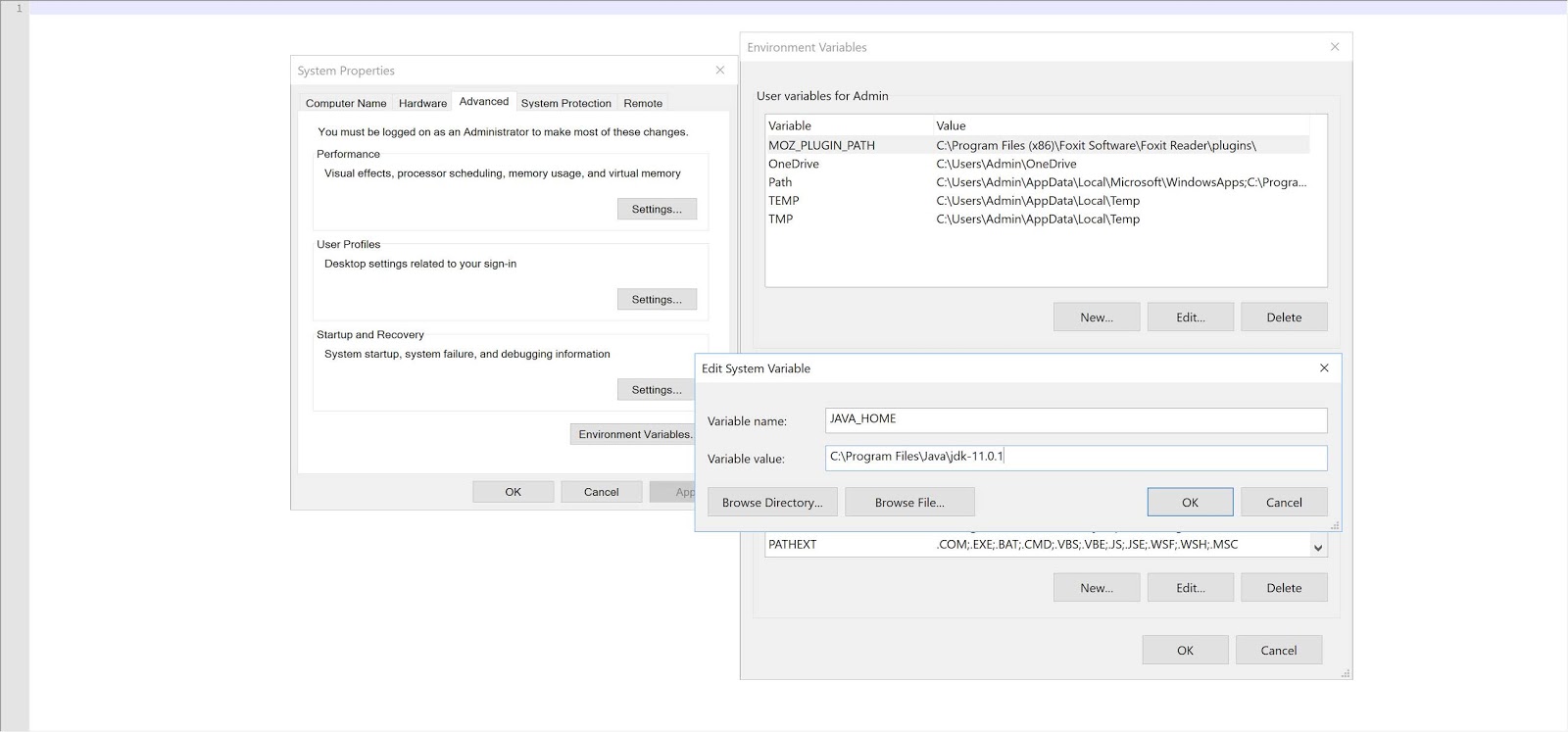
The current version of Java is 11+.

For windows:

You can download JDK11 through <https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>

Then you need to set environment variable in Windows.

* Locate Java installation directory
* Do one of the following:
  + Windows 7 – Right click **My Computer** and select **Properties > Advanced**
  + Windows 8 – Go to **Control Panel > System > Advanced System Settings**
  + Windows 10 – Search for **Environment Variables** then select **Edit the system environment variables**
* Click the **Environment Variables** button
* Under **System Variables**, click **New**
* In the **Variable Name** field, enter *JAVA\_HOME*
* In the **Variable Value** field, enter your JDK installation path
* Choose the **Path** variable, click **Edit**
* In the **Variable** **Value** field, enter *%JAVA\_HOME%\bin* and add it to the end (add ; if there are multiple ones in front)



For Linux:

* Open a terminal window.
* Copy & paste the following into the terminal window and hit Enter. **You may be prompted to enter your password.**

*sudo apt-get install openjdk-11-jdk*

### Install Git

For Windows:

* Download from the following site:

<http://git-scm.com/download/win>

* Open **Powershell** and verify Git is working

*git --version*

For Linux:

* Open a terminal window.
* Copy & paste the following into the terminal window and hit Enter. **You may be prompted to enter your password.**

*sudo apt-get install git*

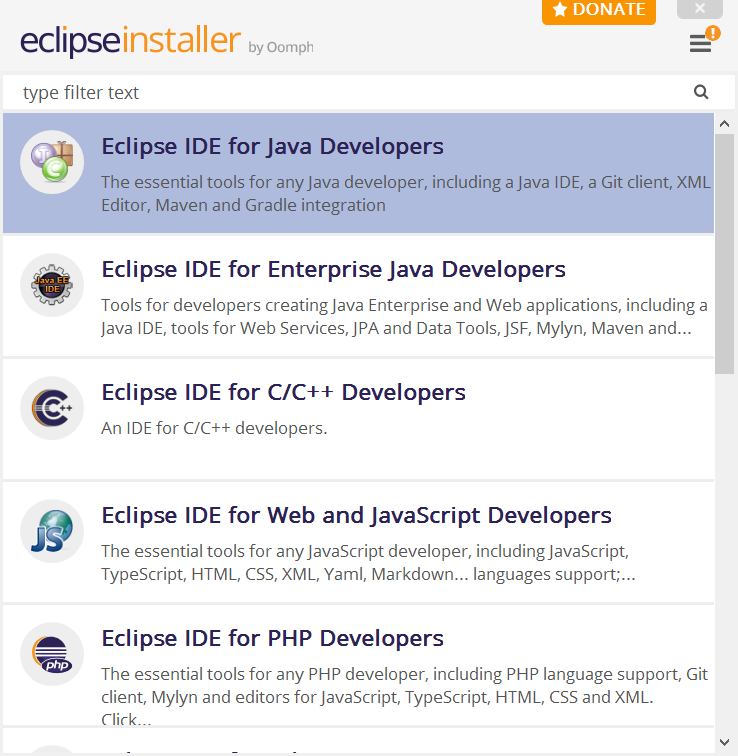
### Install Eclipse

For Windows:

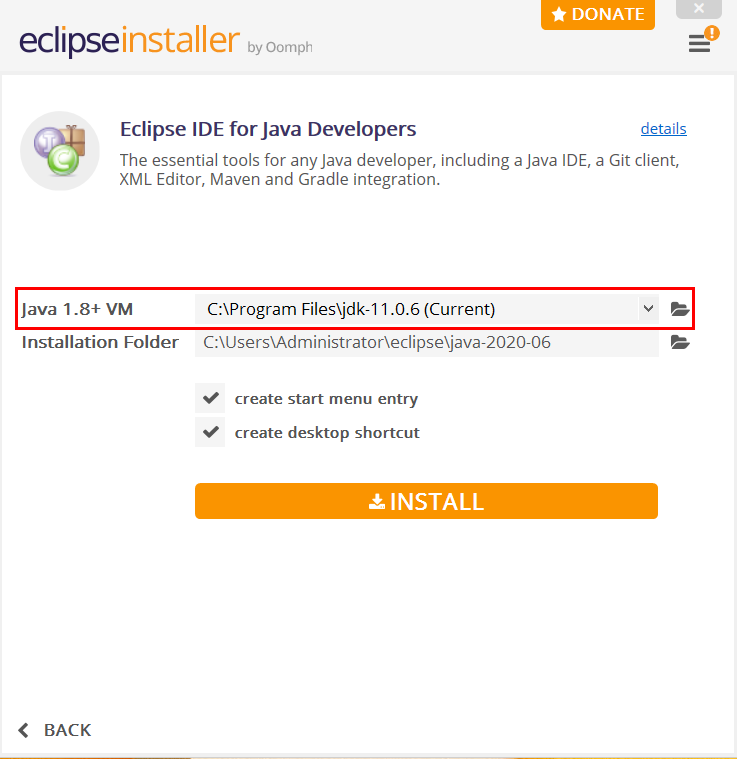
* Download from the following site:

<https://www.eclipse.org/downloads/>

* Choose Eclipse IDE for Java Developers



* Check JDK installation path and install



For Linux:

* Open a terminal window.
* Copy & paste the following into the terminal window and hit Enter. **You may be prompted to enter your password.**

*sudo snap install --classic eclipse*

For Visual Studio Code:

* Download from the following site:

<https://code.visualstudio.com/Download>

### Clone the project

For Windows:

* Open PowerShell
* Copy & paste the following into the terminal window and hit Enter.

*git clone https://gitlab.lip6.fr/mocah/mindmath.git*

For Linux:

* Open a terminal window.
* Copy & paste the following into the terminal window and hit Enter. **You may be prompted to enter your password.**

*sudo git clone https://gitlab.lip6.fr/mocah/mindmath.git*

### Switch the branch

For Windows:

* Go to the root path of the project
* Copy & paste the following into the terminal window and hit Enter.

*git checkout Development*

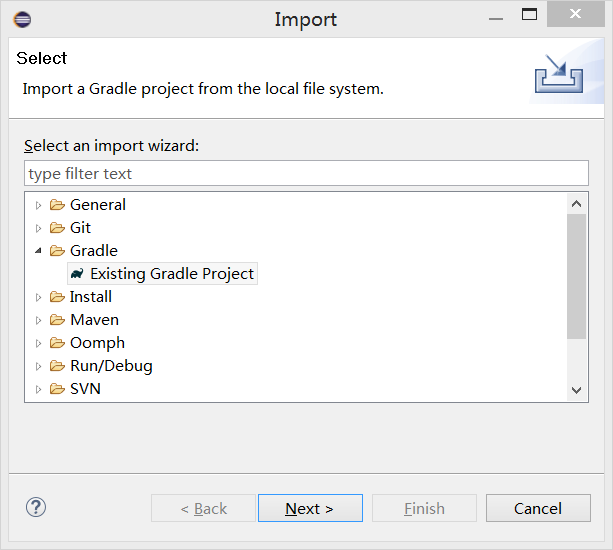
For Linux:

* Go to the root path of the project
* Copy & paste the following into the terminal window and hit Enter. **You may be prompted to enter your password.**

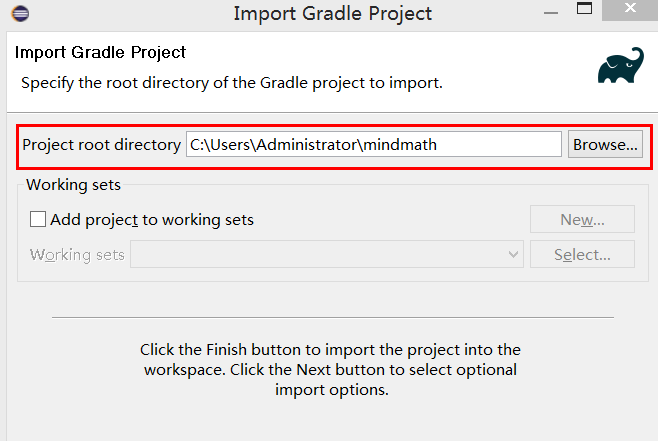
*sudo git checkout Development*

### Import Gradle project with Eclipse

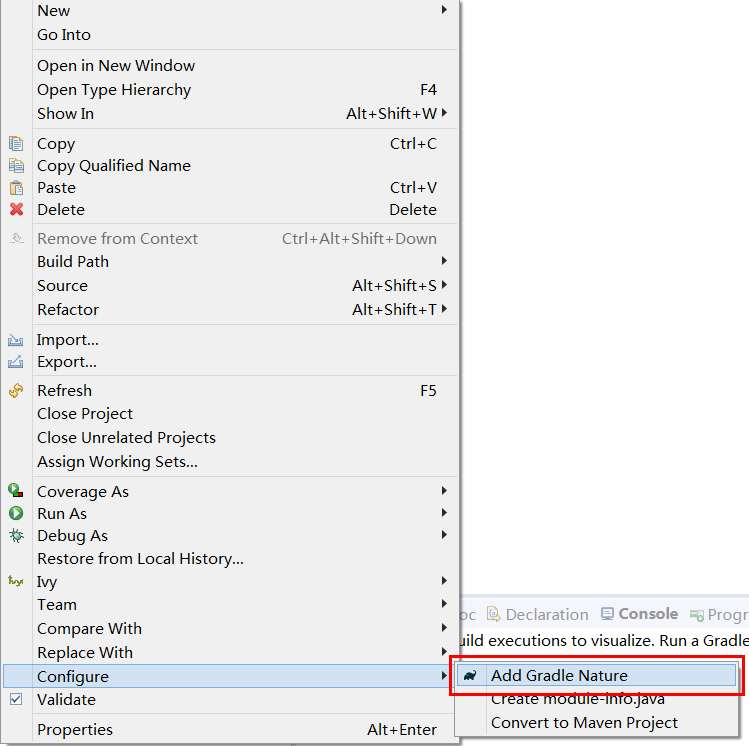
* Open Eclipse
* Click **File** > **Import**, select **Existing Gradle Project**



* Choose the **Project Root Directory** and click **Finish**

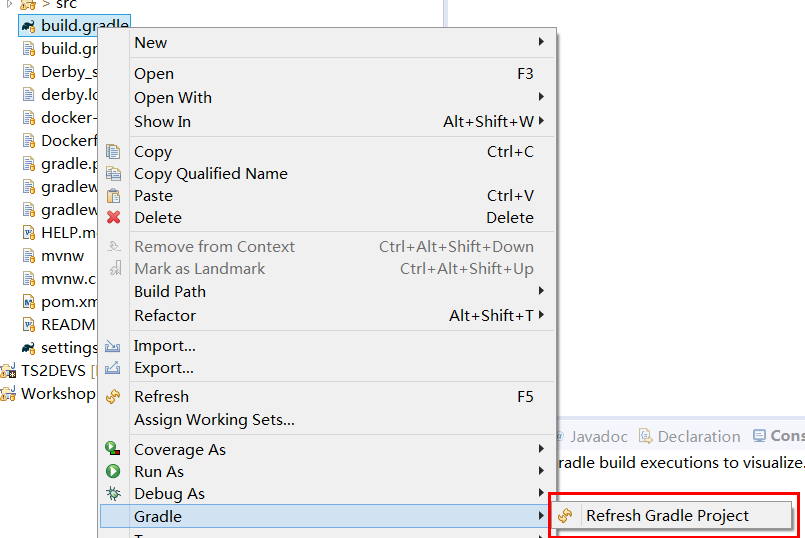


* If the project has already been opened in eclipse, right click the project and choose **Configure** > **Add Gradle Nature**



### Install dependencies via Gradle

Right click the **project** or **build.gradle**, click **Gradle** > **Refresh Gradle Project**



### Run the server at localhost

For Windows:

* Open PowerShell
* Go to the root path of the project
* Copy & paste the following into the terminal window and hit Enter.

*./gradlew bootRun*

For Linux:

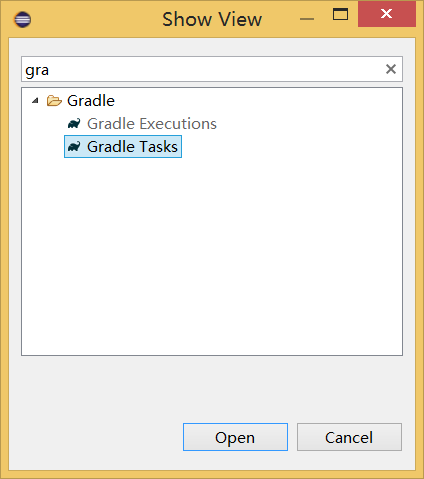
* Go to the root path of the project
* Copy & paste the following into the terminal window and hit Enter. **You may be prompted to enter your password.**

*sudo ./gradlew bootRun*

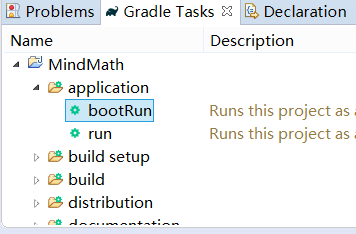
For Eclipse:

* Open **Gradle Tasks** view. Click **Window** > **Show View** > **Other**, select **Gradle Tasks**.

The server is started at <http://localhost:8080/>



* Double click **application** > **bootRun**



### Generate WAR file and deployment

For Windows:

* Open PowerShell
* Go to the root path of the project
* Copy & paste the following into the terminal window and hit Enter.

*./gradlew bootWar*

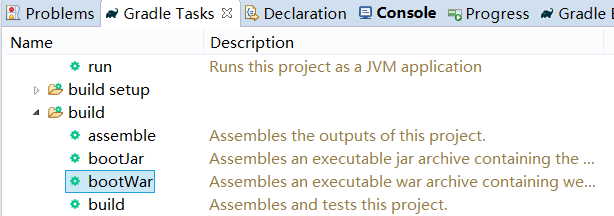
For Linux:

* Go to the root path of the project
* Copy & paste the following into the terminal window and hit Enter. **You may be prompted to enter your password.**

*sudo ./gradlew bootWar*

For Eclipse:

Double click **build** > **bootWar**



The WAR file is in the path ../mindmath/build/libs/MindMath-0.0.2-SNAPSHOT.war

For docker:

* Download Docker via the following site:

<https://www.docker.com/get-started>

* Run docker compose, please check **README** in <https://gitlab.lip6.fr/mocah/mindmath>

For tomcat:

* Download Tomcat via the following site:

<https://tomcat.apache.org/download-90.cgi>

* Put the WAR file in the **../apache-tomcat-9.0.27/webapps**
* Start the tomcat by open **../** **apache-tomcat-9.0.27/bin/startup.bat(.sh)**

Attention: image folder (..\mindmath\src\main\resources\static\images) may cause conflict in Docker compose in version 1.1, delete it before generating WAR file.

### Unit Test

For Windows:

* Open PowerShell
* Go to the root path of the project
* Copy & paste the following into the terminal window and hit Enter.

*./gradlew test*

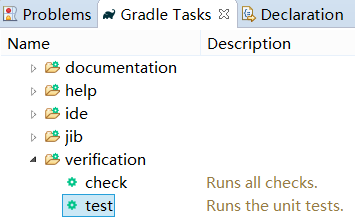
For Linux:

* Go to the root path of the project
* Copy & paste the following into the terminal window and hit Enter. **You may be prompted to enter your password.**

*sudo ./gradlew test*

For Eclipse:

Double click **verification** > **test**



The results are in the path ../mindmath/build/reports/tests/test/index.html

Version 1.1: The simulation engine (Thibaut SF) is inside spring test. The configuration of simulator is in ../mindmath/src/test/java/com/mocha/mindmath/datasimulation/AppConfig.java.

* The version of server
* The expert mode for version 1.0
* The production server selection
* Number of learners per profile
* Random order of profiles
* Iteration times
* The head name for each learner

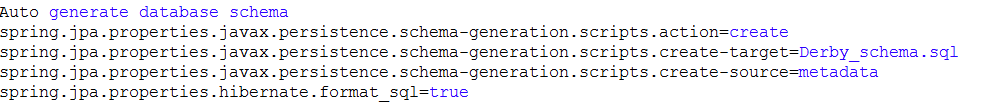
The simulation results are in ../mindmath/build/simulated



### Generate SQL schema

The server can generate database schema for Derby. Just activate the following file

../ mindmath/src/main/resources/application.properties



For the JSON Cabri data schema, please check:

<https://drive.google.com/file/d/1AE5bvcXmrN_Rr8IbTkmlqvG-Rd8z9NVe/view?usp=sharing>

### Update the images in MindMath server

Download FileZilla and install:

<https://filezilla-project.org/>

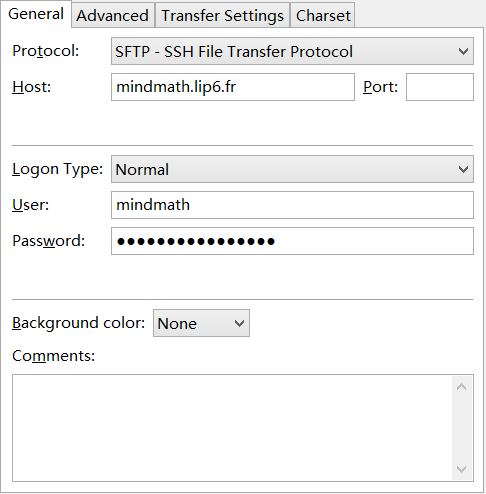
Open FileZilla and then choose site manager.

Create a new site with following information

Host: mindmath.lip6.fr

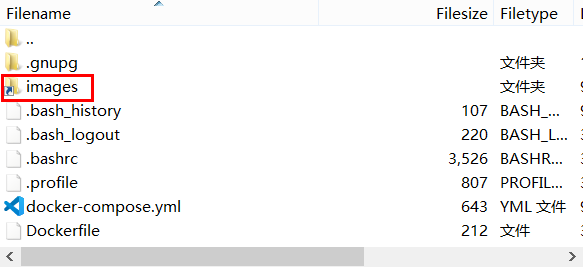
User: mindmath

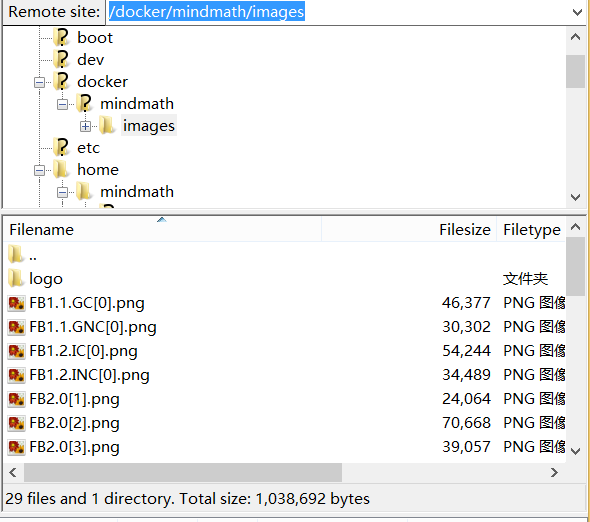
Password: AhPioy4aieX1eeva



Connect the server and go to the path /home/mindmath, here is used to put WAR file and docker-compose.yml for updates.

Choose the folder images, it will direct to the path /docker/mindmath/images, here for images.





## MindMath LIP6 APIs

All the APIs are presented in <https://mindmath.lip6.fr/swagger-ui.html>. You can use this or POSTman (<https://www.postman.com/>) for remote configuration of MindMath server.

Host:

* http://localhost:8080
* https://mindmath.lip6.fr

### Task Controller

Document of transformations between Cabri-LIP6-Domoscio-Tralalere

<https://docs.google.com/document/d/1vLtmdNhD7TQXo2_Yzv9LE6zQ7AQsweo4rvPiB_yOPNI/edit?usp=sharing>

|  |  |  |
| --- | --- | --- |
| **HTTP request** | **Header** | **BasePath** |
| POST | Authorization = mocah | /task |
| POST | Authorization = mocah | /task/test |
| POST | Authorization = mocah | /task/v1.0 |
| POST | Authorization = mocah | /task/v1.1 |

The main controller of the server, receive JSON from Cabri and return back feedbacks.

The default version of post task is v1.1. “/task” is linked to “/task/v1.1”.

Version test: generate feedback and statement xAPI without Qlearning since 05/2020. More details in

<https://drive.google.com/file/d/1SM1RSjR7z26Z2ISKHvCXFPEqpzJyFlT4/view?usp=sharing>

Version 1.0: master student Thibaut SF’s work, use Qlearning and produce simple feedback.

Version 1.1: JSON parser version 1.1. External condition in null task, null activity mode, default maxFB, single generator. Statement xAPI generator version 1.1. Expert Learning. Feedback generator version 1.1 (successScore, closeTask, closeTaskReason).

Conditions of close task is in

<https://docs.google.com/document/d/1RH0_WWa93-e5PE5lai4hDJRN1gZ5_pw-yhZh46O1rLI/edit?usp=sharing>

More details in

<https://drive.google.com/file/d/1a3KGYqfX9Zlq4DVhkdA1zM_KMlesItkD/view?usp=sharing>

|  |  |  |
| --- | --- | --- |
| **HTTP request** | **Header** | **BasePath** |
| GET | Authorization = mocah | /task |
| GET | Authorization = mocah | /task/v1.0 |
| GET | Authorization = mocah | /task/v1.1 |

The default version of post task is v1.1. “/task” is linked to “/task/v1.1”.

Version 1.0: Get all received JSON Cabri from Database Derby. More details in

<https://drive.google.com/file/d/16A-oVWC-7Db8AkRxggMvFcpZylLt5kYg/view?usp=sharing>

Version 1.1: Get both received JSON Cabri, decision from expert learning and feedbacks from Database Derby.

|  |  |  |
| --- | --- | --- |
| **HTTP request** | **Header** | **BasePath** |
| DELETE | Authorization = mocah | /task |
| DELETE | Authorization = mocah | /task/v1.0 |
| DELETE | Authorization = mocah | /task/v1.1 |

The default version of post task is v1.1. “/task” is linked to “/task/v1.1”.

Version 1.0: Delete all received JSON Cabri from Database Derby.

Version 1.1: Delete both received JSON Cabri, decision from expert learning and feedbacks from Database Derby.

### Learning Controller

The policy to select a feedback from generator and family task is based on decision tree.

<https://drive.google.com/file/d/1saYAE4SzvUkoaIJlsC2VfLamu_RqRUMq/view?usp=sharing>

1. Initialize decision tree and Qlearning

|  |  |  |  |
| --- | --- | --- | --- |
| **Request** | **Header** | **Parameter** | **BasePath** |
| POST | Authorization = mocah |  | /learning/init/decisiontree |
| GET | Authorization = mocah |  | /learning/qlearning/qvalues |

The initialization of decision tree is based on local file

.. mindmath/src/main/resources/assets/decisiontree/decisiontree.json

For POST, if request body is “null”, just initialize qlearning and decision tree from local file. If not “null”, according to the format of local file, overwrite local file and initialize qlearning and decision tree.

For Get, return the state of the decision tree and the value of Qlearning.

1. Initialize Expert Learning

|  |  |  |  |
| --- | --- | --- | --- |
| **Request** | **Header** | **Parameter** | **BasePath** |
| POST | Authorization = mocah | id\_learner, id\_task | /learning/init/expertlearning |
| GET | Authorization = mocah | id\_learner, id\_task | /learning/expertlearning/qvalues |

The value of expert learning is retrieved from the value of Qlearning.

For POST, remove the expert learning for learner with id\_learner and task with id\_task in Redis.

For GET, check if there is expert learning for learner with id\_learner and task with id\_task in Redis. If yes, return it. If no, return the current expert learning.

1. Initialize Code Error or Update Code Error

|  |  |  |  |
| --- | --- | --- | --- |
| **Request** | **Header** | **Parameter** | **BasePath** |
| POST | Authorization = mocah |  | /learning/errorcode |
| DELETE | Authorization = mocah |  | /learning/errorcode |

The code errors saved in static in

com/mocha/mindmath/server/entity/feedbackContent/ErrorTypeMap.java

For POST, it will replace the new code errors in JSON objects. Example in DELETE response.

For DELETE, it will initialize the code errors.

1. Initialize Penalty or Update Penalty

|  |  |  |  |
| --- | --- | --- | --- |
| **Request** | **Header** | **Parameter** | **BasePath** |
| POST | Authorization = mocah |  | /learning/penalty |
| DELETE | Authorization = mocah |  | /learning/penalty |

The penalties are saved in static in

com/mocha/mindmath/server/entity/feedback/PenaltyMap.java

For POST, it will replace the new penalties in JSON objects. Example in DELETE response.

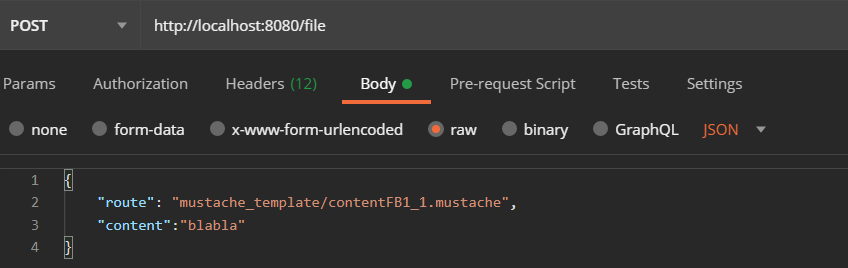
For DELETE, it will initialize the penalties.

### File Controller

|  |  |  |
| --- | --- | --- |
| **HTTP request** | **Header** | **BasePath** |
| POST | Authorization = mocah | /file |
| GET | Authorization = mocah | /file |

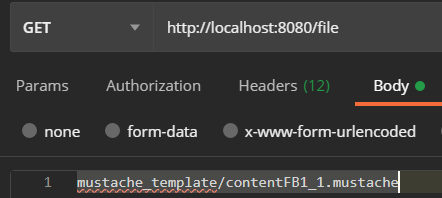
A generic API to control all the files in the folder of resources.

For POST, there is a format of JSON with two parameters “route” and “content”. If not found the file in this “route”, it will return an exception. An example is like this:



Attention, make a backup before this POST.

For GET, put the route directly in the body. An example is like this:



Version 1.1: Use this controller to change the format of html in mustache. The example of the mustache is in .. mindmath/src/main/resources/mustache\_template/contentFB1\_1.mustache

### Feedback Controller

1. Add generator for feedback content, motivation and glossary

|  |  |  |  |
| --- | --- | --- | --- |
| **Request** | **Header** | **Parameter** | **BasePath** |
| POST | Authorization = mocah |  | /feedback/feedbackcontentlist |
| GET | Authorization = mocah |  | /feedback/feedbackcontentlist |
| DELETE | Authorization = mocah | generator | /feedback/feedbackcontentlist |

For POST, add a new generator with feedback content, motivation and glossary. Save this new generator in Derby. Example with format in GET.

For GET, get back all the generators with feedback contents, motivation and glossary from Derby.

For DELETE, delete the received generator with feedback contents, motivation and glossary in Derby.

Attention, please backup before delete. If not, find backups in the following section B-4-b.

1. Update feedback content based on generator

|  |  |  |  |
| --- | --- | --- | --- |
| **Request** | **Header** | **Parameter** | **BasePath** |
| POST | Authorization = mocah | generator | /feedback/feedbackcontent |
| POST | Authorization = mocah | generator | /feedback/motivation |
| POST | Authorization = mocah | generator | /feedback/glossaire |

For POST feedback content, find examples in

.. mindmath/src/main/resources/assets/feedbackcontent.json

This is the main feedback content which get feedback ID leaf, error code from learning, and can get image or modify image path, as well as choose motivation leaf and glossary.

For POST motivation, find examples in

.. mindmath/src/main/resources/assets/motivation.json

Here you can change the content of motivation.

For POST glossary, find examples in

.. mindmath/src/main/resources/assets/glossaire.json

Here you can change the path of glossary or add more glossaries.

### Other APIs:

|  |  |  |
| --- | --- | --- |
| **Request** | **BasePath** | **Description** |
| POST | /file/mustache/imageHTML | Overwrite imageHTML mustache, if backup return the default value Version 1.0 |
| POST | /file/mustache/ videoHTML | Overwrite videoHTML mustache, if backup return the default value Version 1.0 |
| POST | /file/mustache/ generalHTML | Overwrite generalHTML mustache, if backup return the default value Version 1.0 |
| POST | /file/mustache/ glossaireHTML | Overwrite glossaireHTML mustache, if backup return the default value Version 1.0 |
| POST | /learning/test/redis | Posttest values in Redis |
| GET | /learning/test/redis | Get test values in Redis |
| DELETE | /learning/test/redis | Delete test values in Redis |
| POST | /matrix | Test the decision tree class |
| POST | /ontology | Update OWL file in  .. /resources/assets/MindMathOnto.owl  More details in  <https://drive.google.com/file/d/1sDalpjOBl5376JHTlu0GgvWrKJCeULRv/view?usp=sharing> |

## Documentation for LRS

Currently, we have three kinds of statements for each feedback decision. These three statements are designed based on [xAPI specification](https://github.com/adlnet/xAPI-Spec/blob/master/xAPI-Data.md#parttwo). You can find the statement design by the following request. For more information, please go to <https://lrsmocah.lip6.fr/>. More details about post and get statement from MindMath server in

<https://drive.google.com/file/d/1RB-erx9jZSpj3hlrsvCv6iIt1wLk7i-m/view?usp=sharing>

<https://drive.google.com/file/d/1Oqqa2bc70JQdhSA1VGEzS_DO1jTVDFNM/view?usp=sharing>

|  |  |  |
| --- | --- | --- |
| POST | /lrs /test/JXAPIexample | Statement design for LRS |
| POST | /lrs /test/JXAPIexamplePOST | Statement design and post to LRS |
| GET | /lrs /all | Get all statements from store in LRS |
| GET | /lrs/test/all | Get all statements from test store in LRS |
| POST | /lrs/feedbackExperienced | Post Client for experienced statement in LRS |

Statement 1

**Learner:1705 answered https://mindmath.lip6.fr/TaskFamily/ft8.1:**

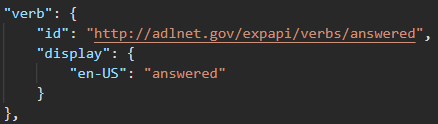
Actor:

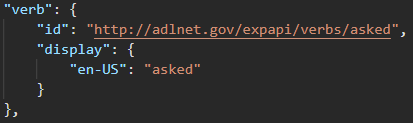
The actor is the learner with ID (1705) and the task with ID (1602624718631) from learning platform Tralalere.



Verb:

There are two verbs “answered” and “asked”. These two verbs are selected based on logs of JSON Cabri.





Object:

The object id is based on task family with its value ft8.1



The interactionType is matching.



This matching has a list of source and a list of target, each one source is linked with one target using correctResponsesPattern.



The logs of JSON cabri is stored in the extensions.



Statement 2

**LIP6 MOCAH responded StatementRef:**

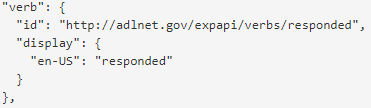
Actor:

The actor is LIP6 MOCAH.



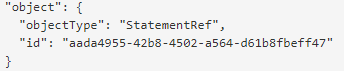
Verb:

The verb is responded.



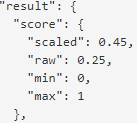
Object:

The object is statement reference which refers to the first statement.



Result:

SuccessScore is in score. Scaled is penalty, raw is the new calculated success score.



Success is linked to correctAnswer. Completion is linked with close task.

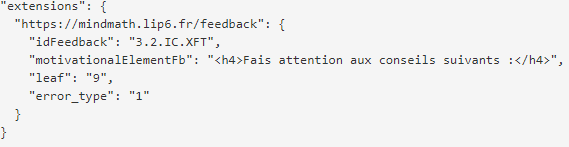


The response is the feedback content which can contain image, video or text.

Version 1.1: feedback content also includes glossary html with new html format in ..mindmath/src/main/resources/mustache\_template/contentFB1\_1.mustache



The feedback ID, motivation, leaf and error\_type are put in the extensions of result.



Statement 3

**Learner:1705 experienced StatementRef**

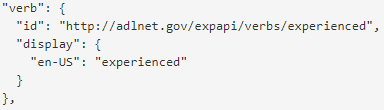
Actor:

The actor is the learner with ID (1705) and the task with ID (1602624718631) from learning platform Tralalere.



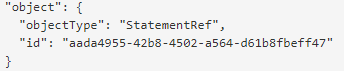
Verb:

The verb is experienced.



Object:

The object is statement reference which refers to the first statement.



Context:

The learner checks the feedback and the path of this feedback is in the extension of context. This context also includes feedbackID, leaf, error code and closeTaskReason (if completion is true).

