MindMath LIP6 Server Manuel

Editor: Yan Wang

Last Modified Date: 10/08/2020

**For the continuous development of MindMath LIP6 server, we need to follow the steps below to construct the IDE:**

1. **Install Java.**
2. **Install Git.**
3. **Install Eclipse (or Visual Studio Code if you want).**
4. **Clone the project into the local machine with Git.**
5. **Switch the branch for production and development.**
6. **Import Gradle project with Eclipse.**
7. **Install dependencies via Gradle.**
8. **Run the server at localhost.**
9. **Generate war file and deploy with Docker.**
10. **Unit Test (optional).**

**At last, we will present the main APIs of MindMath LIP6 server and documentation for LRS.**

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

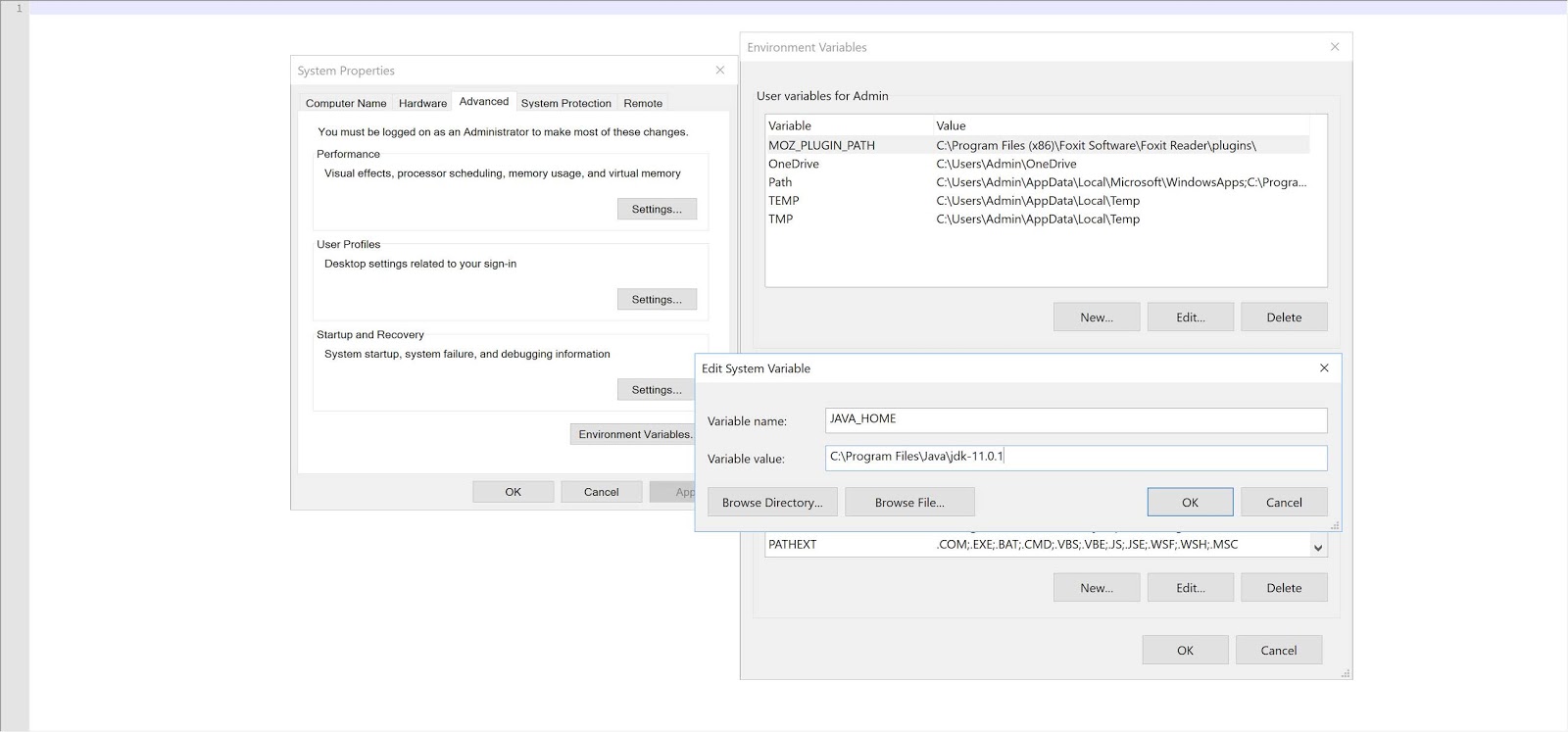
The current version of Java is 11+.

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*

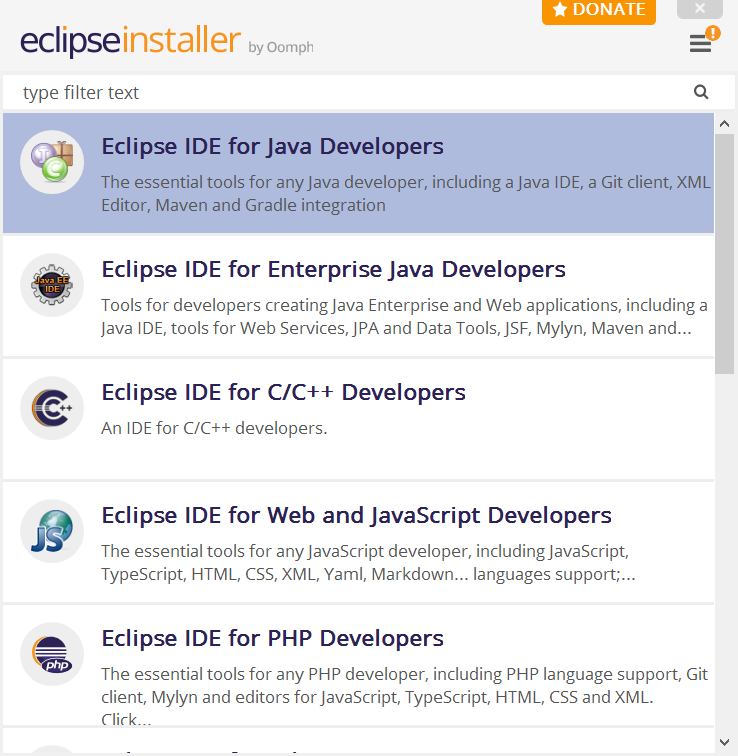
## Install Eclipse

For Windows:

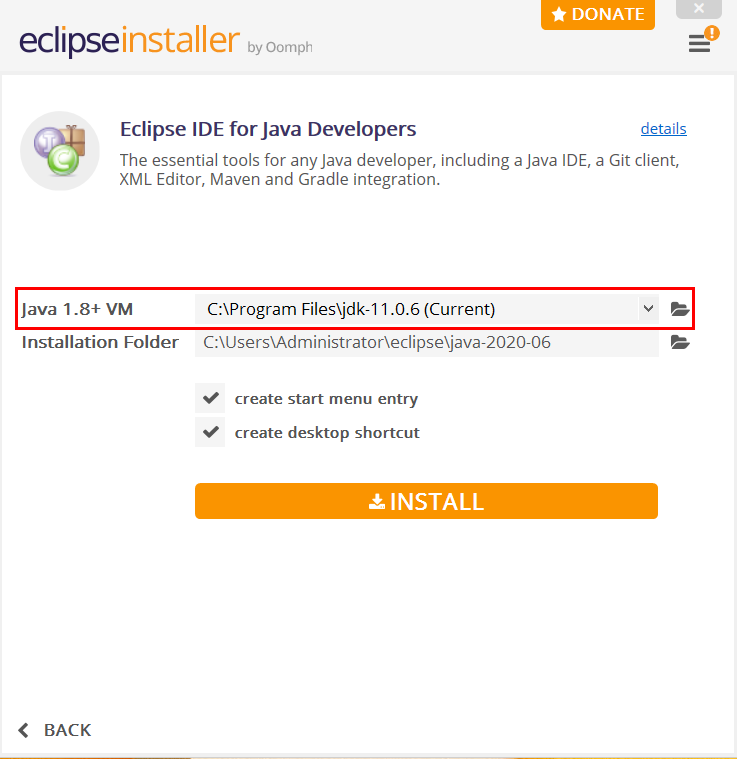
* Download from the following site:

<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*

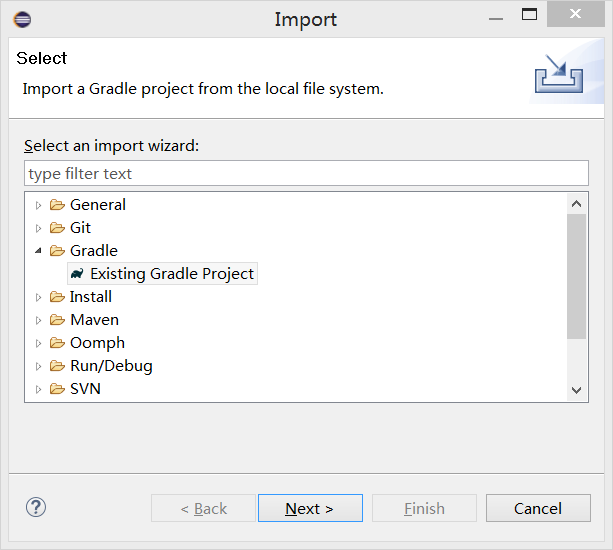
For Linux:

* 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.**

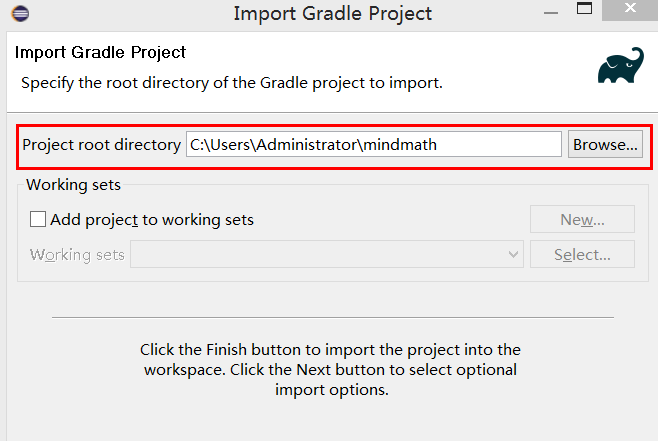
*sudo git checkout Development*

## Import Gradle project with Eclipse

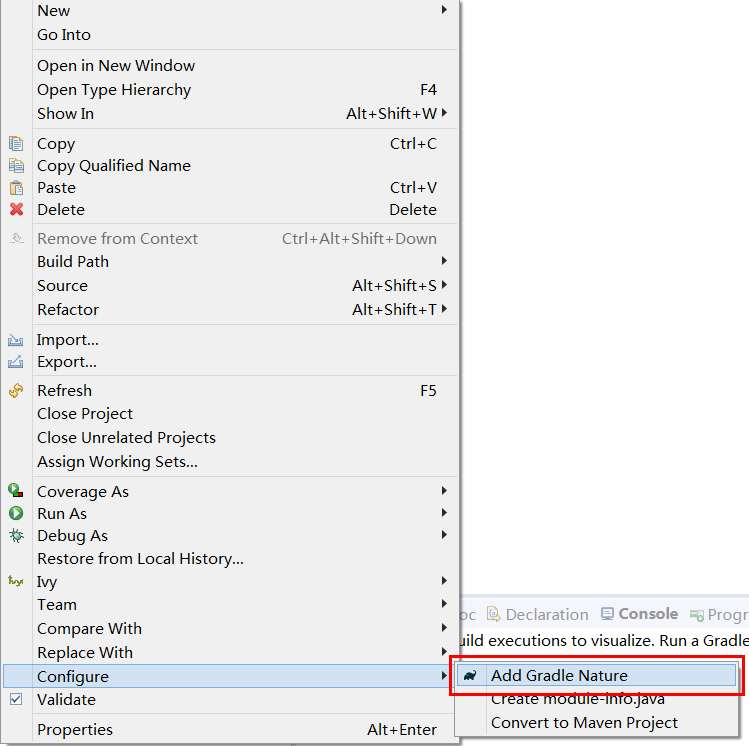
* Open Eclipse
* Click **File** > **Import**, select **Existing Gradle Project**



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

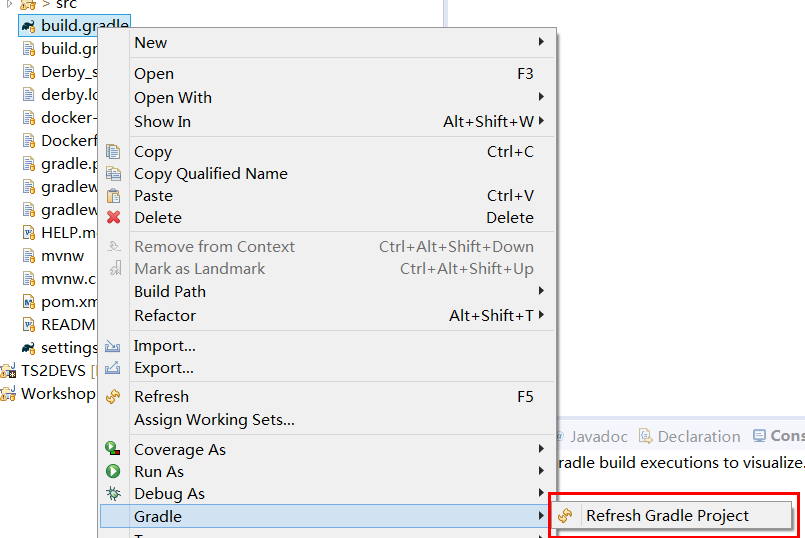


* If the project has already been opened in eclipse, right click the project and choose **Configure** > **Add Gradle Nature**



## 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:

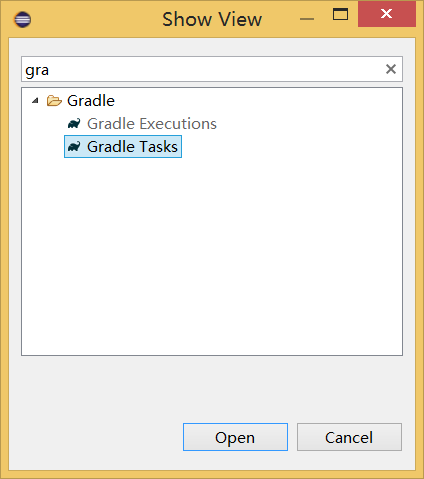
* Go to the root path of the project
* Copy & paste the following into the terminal window and hit Enter. **You may be prompted to enter your password.**

*sudo ./gradlew bootRun*

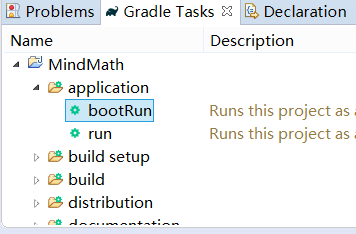
For Eclipse:

* 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*

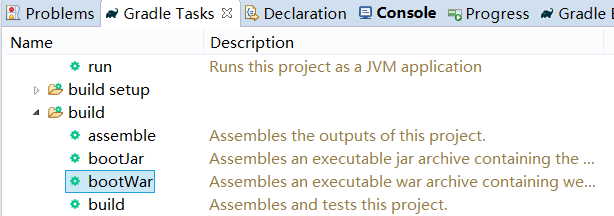
For Linux:

* 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)**

## Unit Test (optional)

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*

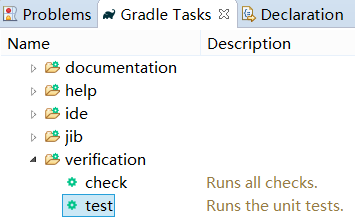
For Linux:

* 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

## MindMath LIP6 APIs

All the APIs are presented in <https://mindmath.lip6.fr/swagger-ui.html>.

Host:

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

BasePath:

|  |  |  |
| --- | --- | --- |
| **HTTP request** | **BasePath** | **Description** |
| POST | /task | Receive JSON Cabri and return back feedback |
| GET | /task | Get all the JSON Cabri from the database |
| GET | /task/qvalues | Get actual qValues from learning |
| GET | /task/test/redis | Get actual qValues from Redis |
| DELETE | /task | Delete all the JSON Cabri in the database |
| POST | /feedback /feedbackcontent | Update feedbackcontent in the database |
| POST | /feedback /motivation | Update motivationin the database |
| POST | /feedback /glossaire | Update glossary in the database |
| POST | /file/mustache/imageHTML | Overwrite imageHTML mustache, if backup return the default value |
| POST | /file/mustache/ videoHTML | Overwrite videoHTML mustache, if backup return the default value |
| POST | /file/mustache/ generalHTML | Overwrite generalHTML mustache, if backup return the default value |
| POST | /file/mustache/ glossaireHTML | Overwrite glossaireHTML mustache, if backup return the default value |
| GET | /file | Read file from local path |
| GET | /lrs/all | Get all statements from LRS |
| GET | /lrs/test/all | Get all statements from test LRS |
| POST | /lrs /test/JXAPI | Generate statement template based on jxapi |
| POST | /lrs /test/JXAPIexample | Statement design for LRS |
| POST | /lrs /test/JXAPIexamplePOST | Statement design and post to LRS |
| POST | /matrix | Update the decision tree class |
| POST | /ontology | Update OWL file |

## Documentation for LRS

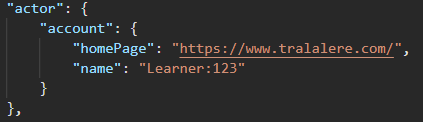
Currently, we have two statements for each feedback decision. These two statements are designed based on [xAPI specification](https://github.com/adlnet/xAPI-Spec/blob/master/xAPI-Data.md#parttwo). You can find the statement design by the following request.

|  |  |  |
| --- | --- | --- |
| POST | /lrs /test/JXAPIexample | Statement design for LRS |
| POST | /lrs /test/JXAPIexamplePOST | Statement design and post to LRS |

Statement 1 **Learner:123 answered https://mindmath.lip6.fr/TaskFamily/ft3.1:**

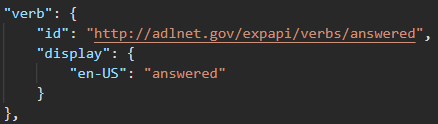
Actor:

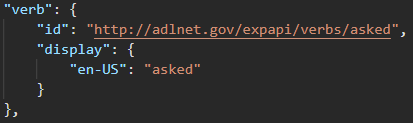
The actor is the learner with ID (123) from learning platform Tralalere.



Verb:

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





Object:

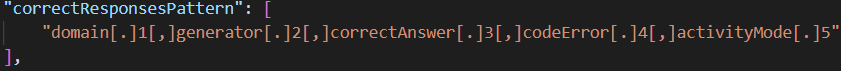
The object id is based on task family with its value ft3.1

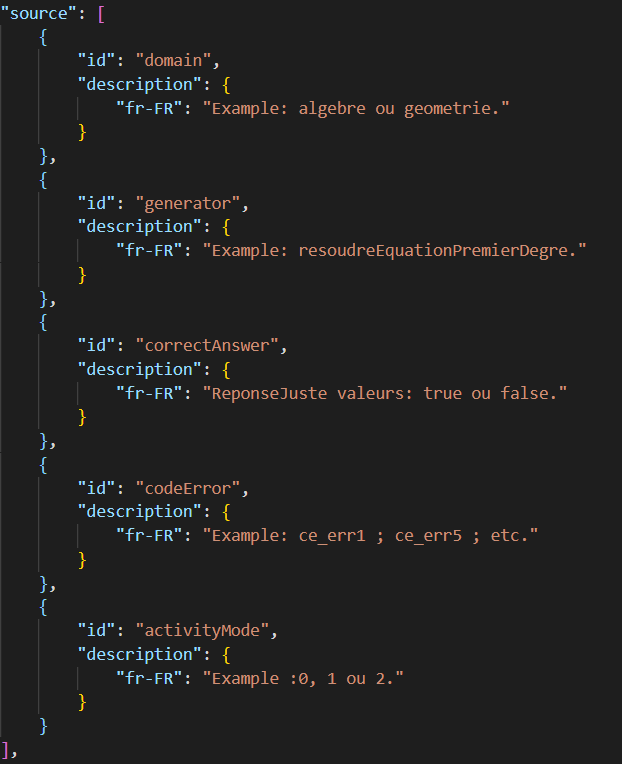


The interactionType is matching.



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







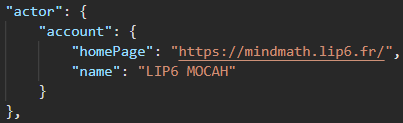
The logs of JSON cabri is stored in the extensions.



Statement 2 **LIP6 MOCAH responded StatementRef (4bc3a97c-d1c3-4f1e-9bf7-98be5a3986f2):**

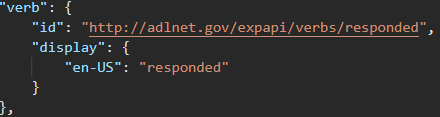
Actor:

The actor is LIP6 MOCAH.



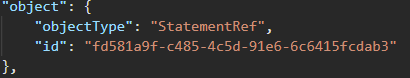
Verb:

The verb is responded.



Object:

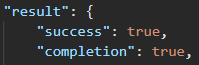
The object is statement reference which refers to the first statement.



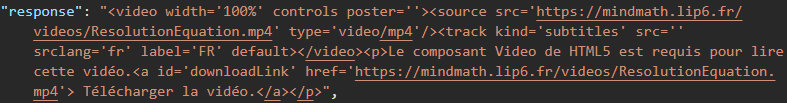
Result:

Success is true if we call decision process, is false if we decide the Cabri JSON is gaming with the system.

Completion is true if decision process work, is false if decision produces an error.



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



The feedback ID, motivation and glossary are put in the extensions of result.

