

EMP (Educational Multiplayer Platform) spec

1. Introduction (scope)

The scope of this document

What is the problem that this solution is trying to solve?

2. Solution structure

A block diagram (functional) of all the systems that interfaces the solution

Description of the structure of the solution

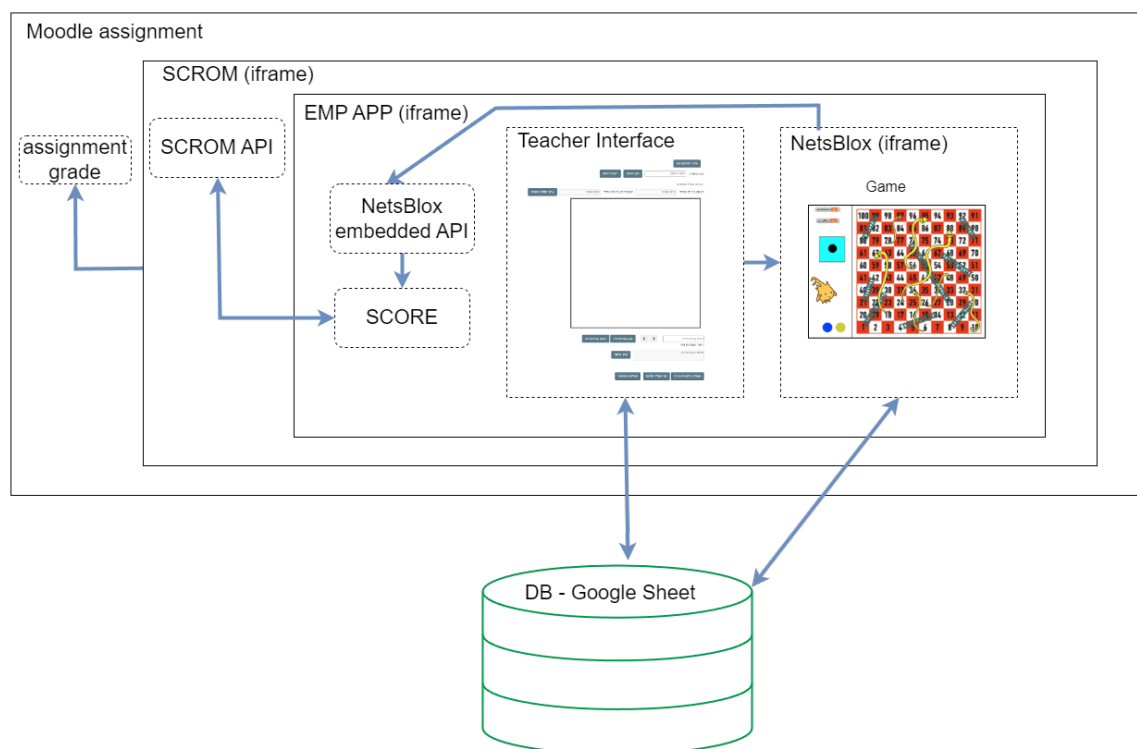


Figure 1: the system logical configuration

3. Functional requirement

*Functional requirements of all the components of the solution
All the scenarios that the solution should support*

3.1. Gamification App

An application which enables loading NetsBlox games and embed them in Moodle. The custom application will be built as a SCORM application and will be loaded into MOODLE via the Scorm activity by which an iFrame will be created . The application will include the following components:

1. SCORM : scorm libraries and configuration XMLs
2. NetsBlox embedded API: a module that will contain the utilities and custom code for interfacing the embedded NetsBlox iFrame.
3. Custom code: will be consist of the following methods:
 - a. remove/loadNetsBlox: removing/loading netsblox game
 - b. addPlayer: assigning player to a role. (TODO: find out if this is possible)
 - c. presentScore: present the score of the current player
 - d. start/stopGame: emulate green flag / red button (TODO: find out if this is possible)
 - e. ...other staff

3.2. Interfaces

3.2.1. MOODLE

3.2.2.

3.2.3. Games - DB

3.3. Supported scenarios

3.3.1. Game creator scenarios

3.3.2. Teacher scenarios

- Teacher loades the the gamification App to a class of his selection in Moodle
- Using the Gamification App teacher loads a NetsBlox Game using the game shared address. Teacher can run the game in two modes:
 - Teacher mode : questions are shown but the answer is always correct
 - Student mode: act in the same way as if the student is running the game

3.3.3. Student scenarios

- Student play the game. He may play with other students.(TODO: find out if a group of students may play a game while another group of students play the same game without mutual interference).
- At the end of the game the score and other performance evaluation are presented to the user.
- User grades are transferred to Moodle

4. Technical solution

The technical specification of all the components

Database structure

Technologies used etc

4.1. User Interface

4.1.1. Teacher UI

Figure 2: Teacher's user interface. The physical proportions may be different in the actual implementation

Description:

- (1) Button to go back to the platform main page
- (2) the name of the game can be used to search the database. when a user starts typing a portion of the name, the records that correlate to the partial notation will be presented for the user to choose from. The software will present results from both the game name fields.
 - (a) "טען משחק" will load the selected game to the game window (4).
 - (b) "הוסף משחק" will pop up a window to add new game to the database.

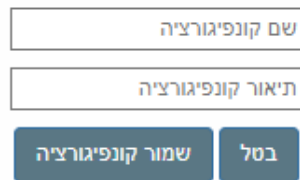
A form with four text input fields and two buttons. The fields are labeled in Hebrew: 'שם המשחק' (Game Name), 'קישור למשחק' (Game Link), 'תיאור המשחק' (Game Description), and 'קישור תמונה למשחק' (Game Image Link). Below the fields are two buttons: 'בטל' (Cancel) and 'הוסף' (Add).

- (3) The default game and configuration for the current assignment in the current course, on first run teacher should setup the default game and configuration that will be load when a student will enter the assignment.
 - (a) "עורך מטלה נוכחית" will pop up a window and let you edit the default game and configuration for the current course and assignment and save it to the database.

A form with two text input fields and two buttons. The first field is labeled 'משחק ברירת מחדל:' (Default Game) and the second field is labeled 'קונפגרציה ברירת מחדל:' (Default Configuration). Below the fields are two buttons: 'שמור' (Save) and 'בטל' (Cancel).

- (4) On-click on טען משחק the game will be loaded in the shown window.
- (5) Choose game configuration: a configuration is the name of the set of questions attached to it. A game may include one or more configurations. When a user enters a partial configuration name, all the corresponding configurations of the loaded game which match the entered text are listed for the user to choose. The user may add a new configuration.
 - (a) "טען קונפגרציה" will load the selected configuration from database (questions, answers, etc.).

(b) "הוסף קונפיגורציה" will pop up a window to add new configuration.



(6) show the description of the current selected configuration, teacher able to add note the a configuration, for example if the teacher write "maths configuration", next time he will add new game or any other teacher is looking for a maths questions for his assignment he will be able to just scroll between the configurations and see the description instead of loading each one and scrolling between the questions to seek if it's a maths related questions.

(a) "ערוך תיאור" will let the teacher edit the selected configuration description.

(7) "שאלות קונפיגורציה" button will show the settings of the current configuration settings, it will open automatically if the "טען קונפיגורציה" button clicked.

(8) "צור שאלה חדשה" on click will pop up a window to add new question to the questions database and also can check the " " button to add it to the current selected configuration.



(9) "שאלות מהמאגר" will pop up a window with all the questions from the database (qa-db sheet), teacher can add question to the current selected configuration by clicking the plus button and also remove the question from the entire database by clicking the "מחק" button, removing a question will remove it from the qa-db and also from all configurations that included this question.



שאלה	מזהה שאלה	סוג שאלה	האספה לקונפיגורציה הנבחרת	מחקר ממסגר השאלות
Q1	1	בחירה מרובה	+	מחקר
Q2	2	בחירה מרובה	+	מחקר
Q3	3	בחירה מרובה	+	מחקר
Q4	4	בחירה מרובה	+	מחקר
Q5	5	בחירה מרובה	+	מחקר

- (10) The user is able to scroll over the questions of the selected configuration by pressing the right or left double-arrows icons. Upon choosing the question the fields of question ID and question title will be updated accordingly.
- (11) "מזהה שאלה" is the question identification, each question will receive unique id up on creation.
- (12) "סוג שאלה" will show the question type, there's two type of questions multiple choice questions or single choice questions, the input will show up this only two choices and will let the teacher select only "בחירה מרובה" or "בחירה יחידה".
- (13) "שאלה נוכחית" it is the current question selected.
- (14) "X תשובה" each input represents the question possible answers, if the question has multiple choices. teacher can check more then one answer for the correct answer.
- (15) each answer have a "feedback" (note) that only the teacher in this stage can see / add.
- (16) The number indicates the earliest stage in which the current question should be displayed. This is relevant only if we are using a configuration with a random order of questions. If during the game a question is randomly selected but the current stage is lower than the earliest stage selected for the question the software will select randomly another question. The default value for this field is 0 (zero) which indicate that the current question may appear in any stage.
- (17) Percentage: the percentage weight of the question. This value is used when computing the final grade of the student (reported to moodle)
- (18) if the questions isn't appear randomly in the game, there's an option to show them by order number.
- (19) If false it indicates that the question may be used just once. For example if we have a random configuration, after that a question was presented to the user it should be removed from the list of possible questions. If true the question will not be removed and may be draw again.
- (20) "מסלול" which path this question related to, for example "geographic", not in use it's preserved for future use.
- (21) "מיומנות" which skills needed for this questions, a field that is preserved for future use.
- (22) "מחקר מקונפיגורציה": the question will be remove from the current configuration. it will not be removed from the qa-db database.
- (23) "בטל": all user's changes will be lost. All fields will return to be empty
- (24) "שמור" will save all changes made by user.

- (25) "מחק ממאגר השאלות : remove from entire database meaning it will remove the question from the qa-db database and also from every configuration that contain this question.

5. Technical Implementation

Stages of implementation

User manual

Instruction of running the system

Instruction for extending the system

(this should include Github libraries and more)

5.1. Github links

Pages: <https://zoharoic.github.io/gamification/>

Repository: <https://github.com/oicenter/gamification>

5.2. The concept (by example):

1. The students are registered to groups in moodle.
2. All students are playing the same game. Each group are playing together
3. The gamification-App is asking the students to enter their group number
4. The hosting application Gamification-app will load the program while transferring the value of addressVariable to the blocks program

Example :

Program name; ConnectViaURL

User (the creator of the program): zsplayer1

Group playing the game: group1

The url of the embedded program will look like this :

<https://editor.netsblox.org/?action=present&Username=zsplayer1&ProjectName=ConnectViaURL&setVariable=addressVariable%3Dgroup1>

Inside the program there will be a created variable by the name: addressVariable having the content: "group1".

All users within group1 will need to add their role id to a list stored in this variable

You may find an example here:

<https://editor.netsblox.org#open:https://raw.githubusercontent.com/NetsBlox/integration-examples/main/embedded/set-variable.xml?setVariable=addressVariable%3Dgroup1>

The html code will look like this

- Please notice that in this example there are 2 containers which will load to instances of the same program side by side. In our implementation we just need one instance.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width" />
    <title>Embedded NetsBlox</title>
    <script charset="utf-8"
src="https://editor.netsblox.org/embedded-api.js"></script>
  </head>
  <body style="margin:0;">
    <iframe id="netsblox" frameborder="0"
style="width:50%;height:100%;position:absolute;"></iframe>
    <iframe id="netsblox2" frameborder="0"
style="left:50%;width:50%;height:100%;position:absolute;"></iframe>
  </body>
  <script charset="utf-8">
    window.onload = () => {
/*
 * This example:
 *   - generates a random name to use to connect to one another
 *   - opens two projects with "addressVariable" set to this value
 *   - uses a cloud variable with the given name to connect to one
another
 */
      const uniqueID = Math.floor(Math.random() * 10000); // This
isn't unique but in practice it should be for this type of project
      const projectURL =
'https://raw.githubusercontent.com/NetsBlox/integration-
examples/main/embedded/set-variable.xml';
      const addressVariable = 'example_' + uniqueID;
      const params = {
        setVariable:
`addressVariable=${addressVariable}`,
      };
      const paramString = Object.entries(params)
        .map(([key, value]) =>
```

```
`${key}=${encodeURIComponent(value)}`)
    .join('&');

    const url =
`https://editor.netsblox.org#open:${projectURL}?${paramString}`;

    const container = document.getElementById('netsblox');
    container.setAttribute('src', url);
    const container2 = document.getElementById('netsblox2');
    container2.setAttribute('src', url);

    console.log('Now, you can interact with netsblox using:',
netsblox);
    console.log('Please open the project found in this directory
to see an example project.');
```

};

</script>

</html>

Or by using iframe

```
<iframe id="netsblox" allowfullscreen= "allowfullscreen"
allow="geolocation; microphone;
camera"src="https://editor.netsblox.org/?action=present&Username=zsplaye
r1&ProjectName=ConnectViaURL&setVariable=addressVariable%3Dgroup1"
frameborder="0" style="width :100%; height:480px;
;position:absolute;"></iframe>
```

Important notice: please notice that this source address activate the game in editor / present mode. The desired mode is embedded for that we need to add “embedMode=true”. Currently the embed mode is missing the green flag and the red button (start/stop program). NetsBlox is suppose to add that. So in the meanwhile we will not use “embed mode”

For more details see the repository <https://github.com/NetsBlox/integration-examples/tree/main/embedded>

```
<iframe id="netsblox" allowfullscreen= "allowfullscreen"
allow="geolocation; microphone;
camera"src="https://editor.netsblox.org/?action=present&Username=zsplaye
r1&ProjectName=ConnectViaURL&setVariable=addressVariable%3Dcode_651"
```

```
frameborder="0" style="width :100%; height:480px;
;position:absolute;"></iframe>
```

Putting all together:

In this example addressVariable = code_651

```
<iframe id="netsblox" allowfullscreen= "allowfullscreen"
allow="geolocation; microphone;
camera"src="https://editor.netsblox.org/?action=present&Username=zsplaye
r1&ProjectName=ConnectViaURL&setVariable=addressVariable%3D:code_651"
frameborder="0" style="width :100%; height:480px;
;position:absolute;"></iframe>
```

this command does the following:

1. create a variable with the name "code_651"(saved in addressVariable)
2. inserts the role public ID in code_651 (each player in the group inserts its own role public id)
3. returns the list of other's role public IDs

code_651

addressVariable

2 myRole@ConnectViaURL@zsplayer1

length: 2

call CloudVariables / getVariable code_651 password

the value of variable code_651

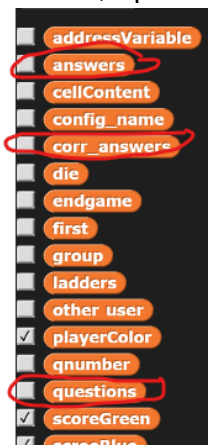
myRole :

send msg message we are connected to item 1 of other user

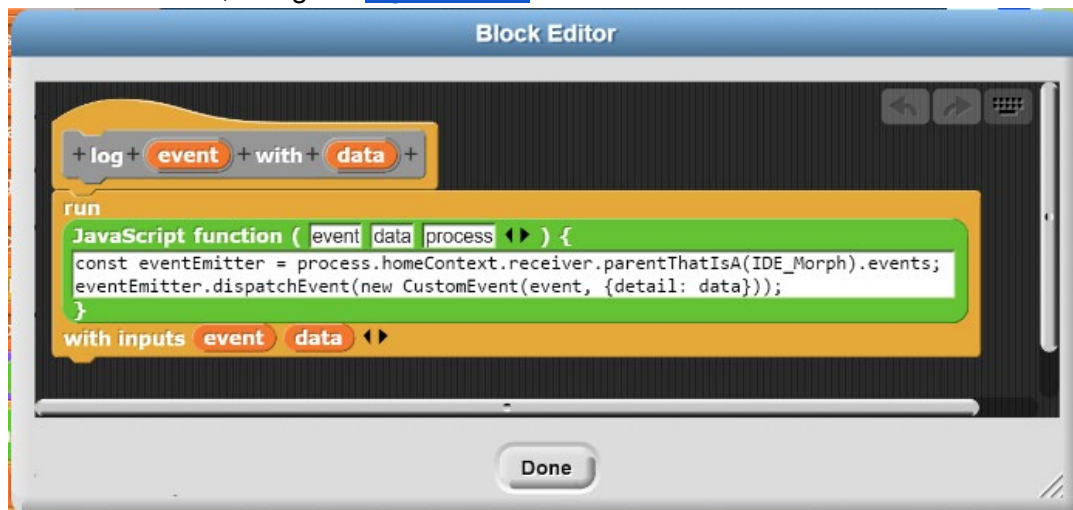
sends message to the first user within the group of users playing the game

Important Blocks for initiating a new game

1. [getDB](#) block is a javascript custom block that pull the questions and answers front he database to the game into 3 variables that should be added "answers" , "correct_ans" , "questions".



- in order to send the score into the SCROM , there's a integration example on [custom event](#) which add the availability to send event to a different CORS , in our case the SCROM domain, using the [log \[\] with \[\]](#) block :



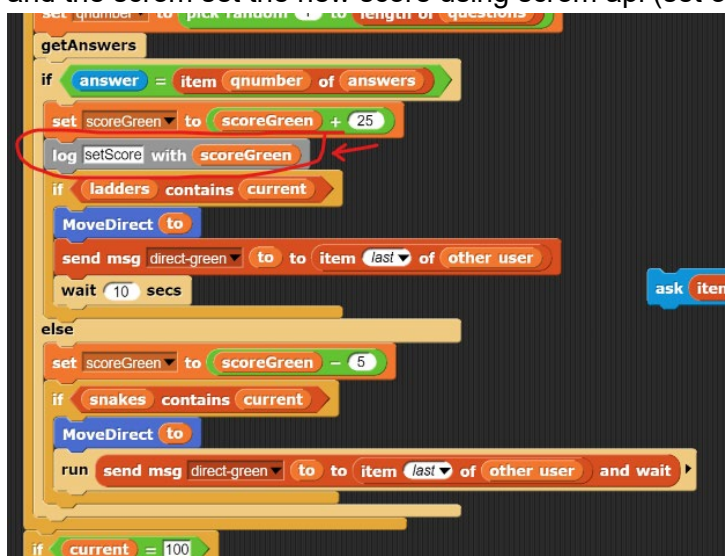
and in the javascript add EventListener
for example in the EMP script.js there's event listener for "setScore":

```
const container = document.getElementById('netsblox');
netsblox = await new EmbeddedNetsBloxAPI(container);
console.log('Now, you can interact with netsblox using:', netsblox);
console.log('Please open the project found in this directory to see an example project.');
```

```
netsblox.addEventListener('setScore', event => {
  console.log(event.detail);
  scorm.set("cmi.score.raw", parseInt(event.detail, 10));
  scorm.save();
});
```

```
loader("OFF");
```

then inside the game if the green player score answer right and his score raised then the log with block will run as shown in the figure below , the new score is sent to the scrom, and the scrom set the new score using scrom api (set on cmi.score.raw)



5.3. Database

Implemented on a google sheet(resembles Microsoft excel) using Google App Script for HTTP requests. Link :

[EMP-DB](#)

Structure:

The database is based on 5 main sheets:

qa-db ▾	games-db ▾	qid-gen ▾	assignments-defconfig ▾	config-description ▾
---------	------------	-----------	-------------------------	----------------------

And another sheet for each configuration.

1) Configurations sheet:

qid - question id, unique for each question.

level - the question level, designed for games with levels.

order - the question order in the game.

weight - question weight, amount of points.

reused - decide whether to show question again in case of previous wrong answer.

path - for future design.

skill - for future design.

qid	level	order	weight	reused	path	skill
1	l1	5	1	TRUE	p1	none
2	l1	2	1	FALSE	p1	none
3	l2	3	1	FALSE	p2	none
4	level_1	4	weight_1	FALSE	path_1	skill_1
5	l4	5	25	TRUE	p7	math
7	l2	6	1	FALSE	p2	none

2) qa-db: normalized form, 4 rows per question.

qid - question id.

question - question content.

answer - answer content.

correct - TRUE if answer is correct, otherwise FALSE.

type - question type, currently there are 2 types: single choice and multiple choice.

feedback - feedback on the question/answer.

qid	question	answer	correct	type	feedback
1	Q1	A1_0	FALSE	בחירה פתוחה	F1_0
1	Q1	A1_1	FALSE	בחירה פתוחה	F1_1
1	Q1	A1_2	TRUE	בחירה פתוחה	F1_2
1	Q1	A1_3	FALSE	בחירה פתוחה	F1_3
2	Q2	A2_0	FALSE	בחירה פתוחה	F2_0
2	Q2	A2_1	FALSE	בחירה פתוחה	F2_1
2	Q2	A2_2	FALSE	בחירה פתוחה	F2_2
2	Q2	A2_3	TRUE	בחירה פתוחה	F2_3
3	Q3	A3_0	FALSE	בחירה פתוחה	F3_0
3	Q3	A3_1	FALSE	בחירה פתוחה	F3_1
3	Q3	A3_2	TRUE	בחירה פתוחה	F3_2
3	Q3	A3_3	FALSE	בחירה פתוחה	F3_3
4	Q4	A4_0	FALSE	בחירה פתוחה	F4_0
4	Q4	A4_1	FALSE	בחירה פתוחה	F4_1
4	Q4	A4_2	TRUE	בחירה פתוחה	F4_2
4	Q4	A4_3	FALSE	בחירה פתוחה	F4_3

3) games-db: contains the games on the platform.

game_name - the game name.

url - link to a netblox game with shared access.

description - description of the game.

img_url

game_name	url	description	img_url
S&L	https://editor.netblox.org/?action=present&Username=rony&id=1&ProjectName=S&L	Snakes And Ladders	https://i.ibb.co/7QpTc7k/Board-hires.png
Game_PH1	https://PLACEHOLDER.COM	place_holder	https://i.ibb.co/ngs9kTX/placeholder.png
Game_PH2	https://PLACEHOLDER.COM	place_holder	https://i.ibb.co/ngs9kTX/placeholder.png
Game_PH3	https://PLACEHOLDER.COM	place_holder	https://i.ibb.co/ngs9kTX/placeholder.png

- 4) qid_gen: contains the next qid for a new question.
- 5) assignment-defconfig: contains default settings for different course assignments.
Using this sheet the platform knows which game to load and which configuration for a certain assignment.

cid - course id, every course in moodle has a unique id.

assigid - assignment id, every assignment in moodle has a unique id.

game_name - game name used for the course assignment.

default_config - configuration name used for the course assignment.

cid	assigid	game_name	default_config
22087	875964	S&L	laddersRops-config1
22087	888760	S&L	game5-config

- 6) config-description: contains descriptions for configurations, shown and can be edited in teacher UI.

config_name - configuration name.

description - configuration description.

config_name	description
laddersRops-config1	חשבון
snakesAndLadder-2022	אנגלית
micTest111	blah
game5-config	as

5.3.1. Google App Script

the database google sheet have a backend "app script", using this script the EMP platform can send query requests to that database inside the sheets.

there's a function called doGet() which is a built in function from app script, that will run when an http request is sent to the published web app link inside the app script.

the function gets one variable "req", the req variable is the http request header.

the header should have up to 3 parameters (can add more in the future), the structure is:

first parameter: the function that should be run;

second parameter: the function parameter one (if needed)

third parameter: a function parameter two (if needed)

the selection which function should run is implemented using switch case see figure:

```

function doGet(req){
  var p1 = req.parameter.param1;
  var p2 = req.parameter.param2;
  var p3 = req.parameter.param3;
  switch (p1) {
    case "getRowsByValues":
      return ContentService.createTextOutput(JSON.stringify(getRowsByValues('qa-db',p2))).setMimeType(ContentService.MimeType.JSON);
    case "getDataFromsheet":
      return ContentService.createTextOutput(JSON.stringify(getDataFromsheet(p2))).setMimeType(ContentService.MimeType.JSON);
      break;
    case "getConfigsName":
      return ContentService.createTextOutput(JSON.stringify(getConfigsName())).setMimeType(ContentService.MimeType.JSON);
      break;
    case "deleteFromDB":
      return ContentService.createTextOutput(JSON.stringify(deleteFromDB(p2))).setMimeType(ContentService.MimeType.JSON);
      break;
    case "deleteFromConfig":
      return ContentService.createTextOutput(JSON.stringify(deleteFromConfig(p2,p3))).setMimeType(ContentService.MimeType.JSON);
      break;
    case "addQuestion_qaDB":
      return ContentService.createTextOutput(JSON.stringify(addQuestion_qaDB(p2))).setMimeType(ContentService.MimeType.JSON);
      break;
    case "addQuestion_config":
      return ContentService.createTextOutput(JSON.stringify(addQuestion_config(p2,p3))).setMimeType(ContentService.MimeType.JSON);
      break;
  }
}

```

doGet return http respond to the sender using this command for example:

```

case "getDataFromsheet":
  return
ContentService.createTextOutput(JSON.stringify(getDataFromsheet(p2))).setMimeType(ContentService.MimeType.JSON);

```

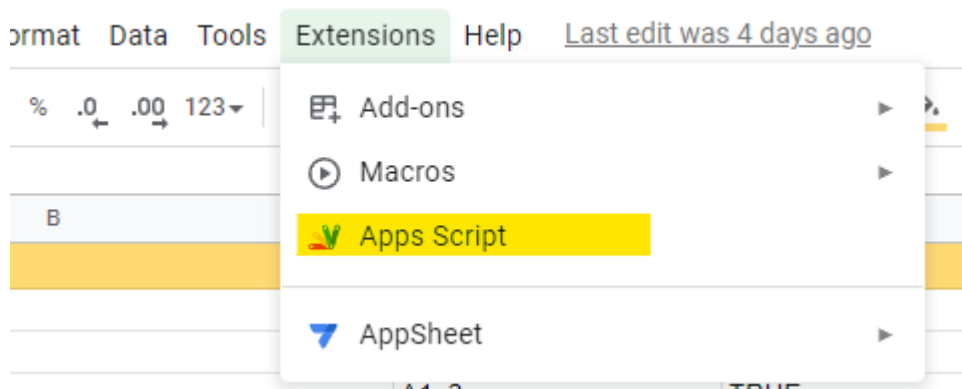
this will run a function called getDataFromSheet (which returns data from a sheet name inside the "p2" variable then it will convert it to a json structure and send it as response using ContentServices.CreatTextOutput()).

Editing the app script:

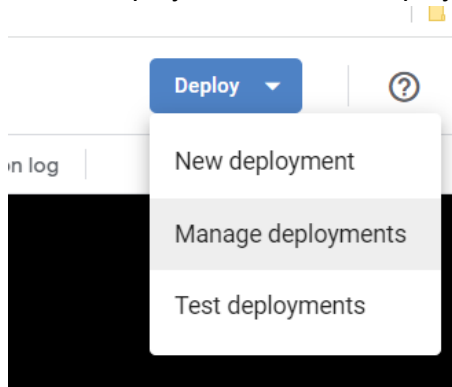
the enter the app script first got to that database google sheet link:


[EMP-DB](#)

then go to Extensions > Apps Script



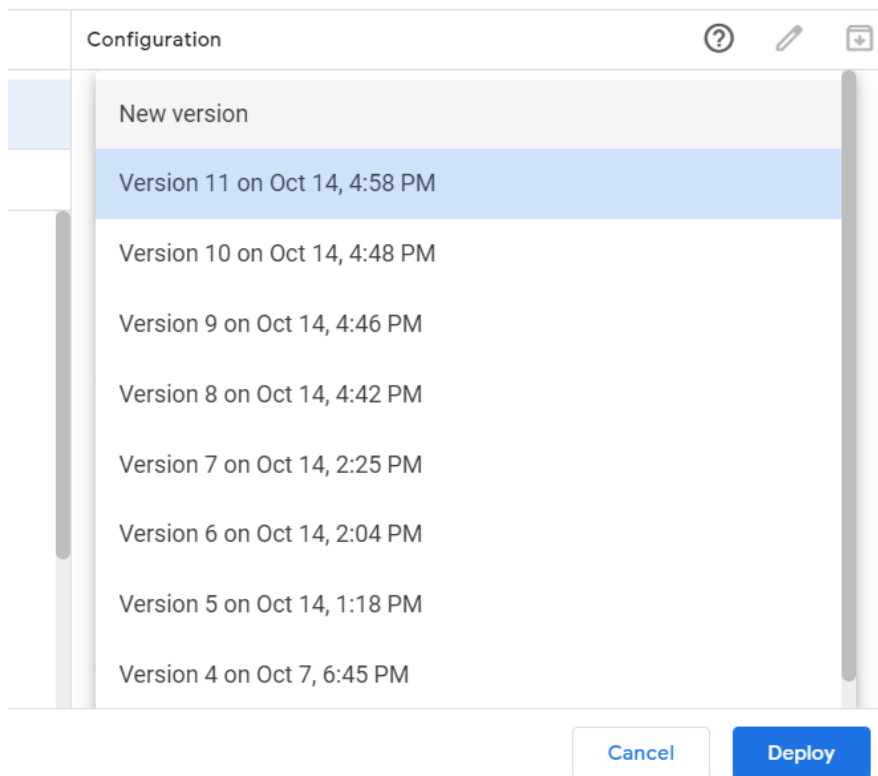
in order to save the changes if made on the app script code you should make a new version for the deployment click on Deploy > Manage deployments



and then click the pencil icon  click the down arrow on the version box



then choose “New Version” , and click “Deploy” button
ments



the web app link is here:

Web app

URL

https://script.google.com/macros/s/AKfycbyH1nxVtvqc_tmss5H5sIMEuGF2Eqh1toN...

 Copy

all functions in the app script are implemented using the [Google Sheet API \(APP SCRIPT\)](#).