Documentation for the REST API Implementation for Labvanced.com

Document version: version 1.0.3
Document last changed on: 2020-11-17

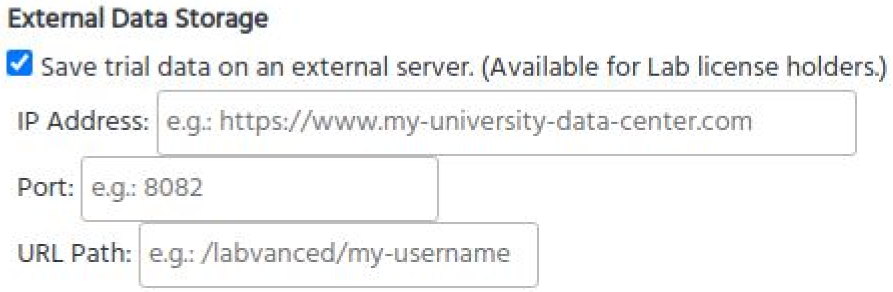
Usage: The main use case for the Labvanced REST API is to transfer participant data in “real time” to an external / remote server, instead of hosting the recorded data on the Labvanced servers.

PLEASE NOTE: Contrary to many API use cases, what is described here is NOT how to call the Labvanced API / endpoints (such a functionality does not exist for now), but rather how you must configure/implement your backend, such that our platform will be able to automatically call the endpoints you have implemented to sent the participants data directly to your backend/database instead of ours.

Availability: This functionality is only available to Lab license holders.

1. Setup in Labvanced

To use the REST API, one first has to activate the “external data storage” option in “study settings” of the respective Labvanced study (see screenshot):



External Data Storage

☑ Save trial data on an external server. (Available for Lab license holders.)

IP Address: e.g.: https://www.my-university-data-center.com

Port: e.g.: 8082

URL Path: e.g.: /labvanced/my-username

1.1 Parameters:
IP Address: Enter your static IP address where your external server is running / publicly *available.*
Port: The port which should be used to send the data

URL Path: The URL path which should be appended after the port. This is particularly useful to “namespace” your projects / incoming data, such that data from different experiments are sent to a different URL path. The URL path is optional, but we strongly recommend to use it.

2. Create the server / script for receiving the data

Based on your settings in step 1, you have to implement a REST server which is available at the specified IP address, port, and url path. A simplified version / example of such a server is available on our Github page at:
https://github.com/Labvanced/python_REST_API_example/tree/main

2.1 Data format: All data is sent in JSON format, so make sure you parse the incoming data correctly as JSON.

2.2 Request Types: All request types are POST requests as they are used to send data out. No return values are required.

2.3 Types of requests / routes: Each time a participant takes part in a study, in total 8 different *types* of POST requests will be sent to the (external /your) server. Each type of POST request has a different route (url path), which will be appended to the specified URL path in step 1. Also each type of request has a different structure/payload, which should be accounted for when parsing and storing the data. See details in section 3.

3. API endpoint definitions / routes:

In the following, you will see a detailed explanation of each route / endpoint that is used/called when a participant takes part in the experiment. Ideally we suggest that you implement all of these endpoints (besides 3.1), and make sure all the data is appropriately parsed and stored in your database.

Which endpoints are used, which server is called when?

Endpoint 3.1 “/startExpPlayer” is only sent to the Labvanced server to load the experiment definition/JSON file and associated stimuli from our server.

Endpoints 3.2, 3.3, 3.4, 3.7, and 3.8 are sent both to the Labvanced server (needed for group balancing and general experiment flow) and the external/your server.

Endpoints 3.5 and 3.6 are only sent to the externla/your server. Please note all trial/participant data is sent via route 3.6, which therefore is the most important endpoint to implement.

3.1 Loading of the experiment

Route: /startExpPlayer
Type: POST
Called when: The participant visits a certain experiment URL on labvanced.com
Main function on Lavanced server: Initializes the loading of the experiment.
Payload:

Field name	**description**	**data type**	**required**
expId	Unique identifier for the study	integer	yes
isTestrun	Whether data is recorded or not	boolean	no
subject_code	Identifier of the subject via url param	JSON	no
token	A unique identifier for participants with longitudinal studies	Token identifier	no
askSubjData	Whether to show an initial survey to used to put subjects into different groups	boolean	yes

Note: Does not need to be implemented on external servers, as this only initializes the loading of the experiment from the Labvanced server to the client computer

3.2 Group and session selection

route: /startFirstPlayerSession
type: POST
Called when: Route 3.1 " /startExpPlayer" returns successfully from the Labvanced server.
Main function on Lavanced server: Used to assign group and session numbers to a subject (server side balancing)
payload: JSON with the following fields:

Field name	description	data type	required
expId	Unique identifier for the study	integer	yes
expSessionNr	Unique identifier for the recording session for this participant (counting upwards)	string	yes
subject_code	Identifier of the subject via url param	string	no
token	A unique identifier for participants with longitudinal studies	Token identifier	no
survey_data	JSON of the pre-study questionnaire, fields described below.	JSON	no
survey_data.selectedGender	Selected gender of the participant	string	no
survey_data.selectedAge	Selected age of the participant	integer	no
survey_data.selectedCountry	Selected country/location of the participant	string	no
survey_data.selectedLanguage	Selected (first) language of the participant	string	no
isTestrun	Whether data is recorded or not	boolean	yes
runOnlyGroupNr	Only used to test run a certain group number without data recordings	Integer | false	no
runOnlySessionNr	Only used to test run a certain session number without data recordings	Integer | false	no
groupNr	Group number for this subject	integer	yes

sessionNr	Session number in a longitudinal study	integer	yes
group_name	The name of the group for this subject	string	yes
session_name	The name of the session for this subject	string	yes
experiment_name	The experiment name (not the publication ame)	string	yes

3.3 Start of the experiment

route: /setPlayerSessionStartedTime

type: POST

Called when: The participant presses “Start Experiment”

Function on Lavanced Server: Saves the start time, session and group name of the experiment and starts the experiment.

Payload:

Field name	description	data type	required
expId	Unique identifier for the study	integer	yes
expSessionNr	Unique identifier for the recording session for this participant (counting upwards)	string	yes
start_time	Start time of the experiment	UNIX timestamp	yes
sessionNr	Session number in a longitudinal study	integer	yes
group Nr	Group number for this subject	integer	yes
token	A unique identifier for participants with longitudinal studies	Token identifier	A unique identifier for participants with longitudinal studies

3.4 Add meta data information

route: /addMetaInfo

type: POST

Called when: The participant presses “Start Experiment”

Function on Lavanced Server: Saves the meta information on the server.

Payload:

Field name	description	data type	required
expId	Unique identifier for the study	integer	yes
expSessionNr	Unique identifier for the recording session for this participant (counting upwards)	string	yes
var_data	The JSON holding the meta information	JSON	yes
var_data.browserSpec	The browser which is used by the subject	string	no
var_data.versionSpec	The browser version which is used by the subject	string	no
var_data.systemSpec	The device type / OS which is used by the subject	string	no
var_data.agentSpec	The complete user agent string	string	no
var_data.fullscreen	Indicates If the study is always in fullscreen	boolean	no
var_data.timeDelayMean	The Javascript callback precision mean offset in milliseconds	float	no
var_data.timeDelayStd	The Javascript callback precision standard deviation in milliseconds	float	no
var_data.crowdsourcingCode	The crowdsourcing / completion code for subjects	string	no
var_data.crowdsourcinSubjId	The crowdsourcing / worker id of the subject	string	no
var_data.subjCounterGlobal	A global subject counter for number of subject in the study	integer	no
var_data.subjCounterPerGroup	A subject counter per group in the study	Array of integers	no

var_data.roleId	The unique role id for multi-user studies of the subject	integer	no
var_data.multiUserGroupId	The global unique multi-user group id for multi-user studies	uuid	no
var_data.displayedLanguage	The selected display language for multi-language studies	string	no
var_data.pixelDensityPerMM	The pixel density of the screen	float	no
var_data.screenHeight	The screens’ height in pixel	integer	no
var_data.screenWidth	The screens’ width in pixel	integer	no
var_data.windowHeight	The window /viewport height in pixel	integer	no
var_data.windowWidth	The window/viewport width in pixel	integer	no

3.5 Record information about a task

route: /recordStartTask
type: POST
Called when: A new tasks starts in the experiment flow
Function on Lavanced Server: Not called when external API requests are activated
Payload:

Field name	description	data type	required
expId	Unique identifier for the study	integer	yes
expSessionNr	Unique identifier for the recording session for this participant (counting upwards)	string	yes
blockNr	Current block number (increasing counter)	integer	yes
blockId	Unique Id ob the current block	uuid	yes
blockName	Name of the block	string	yes
taskNr	Current task number (increasing counter)	integer	yes
taskId	Id of the current task as defined in the editor (same between subjects)	uuid	yes
recTaskId	Server generated Id for the current recording	integer	yes

	task (different between subjects)		
taskName	Name of the current task	string	yes
start_time	Start time of the task	UNIX timestamp	yes

3.6 Record trial data

route: /recordTrial

type: POST

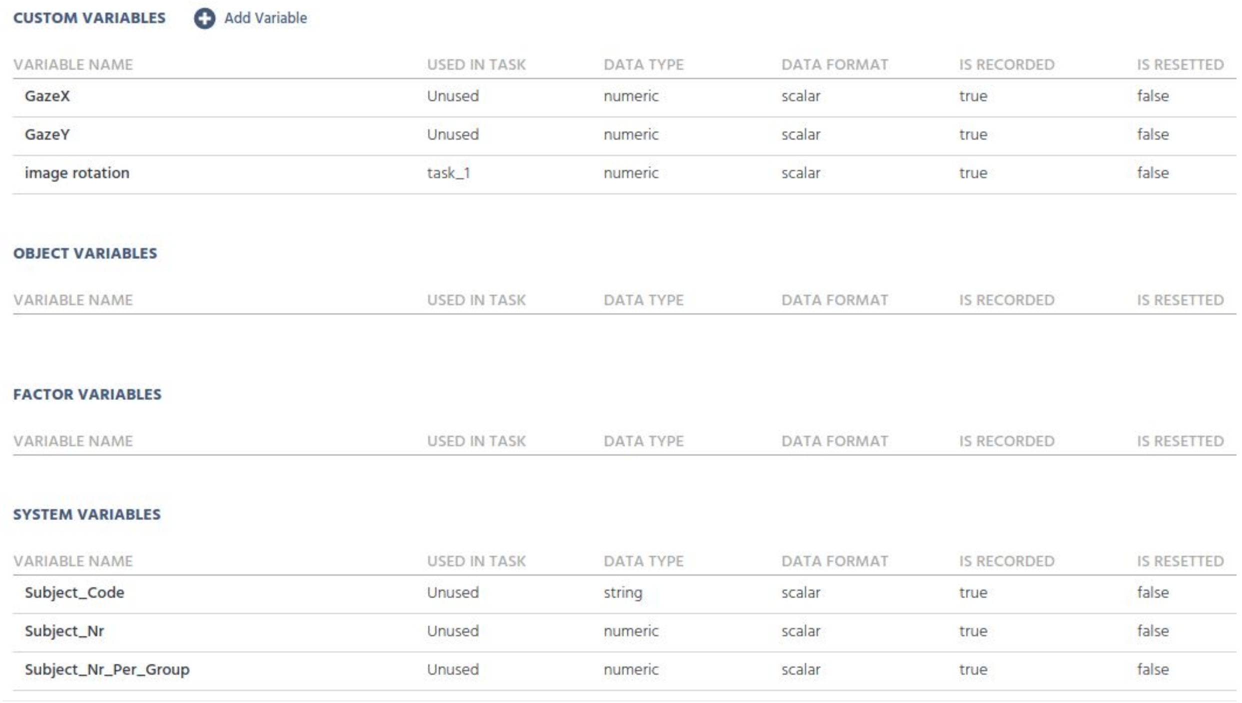
Called when: A trial is finished OR a custom record action is executed

Function on Lavanced Server: Not called when external API requests are activated.

Payload:

Field name	**description**	**data type**	**required**
expId	Unique identifier for the study	integer	yes
expSessionNr	Unique identifier for the recording session for this participant (counting upwards)	string	yes
recTaskId	Identifier for the current task	integer	yes
taskId	Id of the current task as defined in the editor (same between subjects)	uuid	yes
trialNr	Trial Number	integer	yes
recData	The main data / all user created variables per trial are stored here.	JSON	yes

Note: The exact data which is recorded per trial depends on the specific experiment and task. An overview for each study is available under the “Variables” tab of the specific experiment (see screenshot). Also please note that the external API variable names will be used as a key in the JSON structure. So please make sure that variable names are unique. Otherwise you would overwrite data. The Labvanced system usually enforces the user to use unique variable names.



CUSTOM VARIABLES

VARIABLE NAME	USED IN TASK	DATA TYPE	DATA FORMAT	IS RECORDED	IS RESETTED
GazeX	Unused	numeric	scalar	true	false
GazeY	Unused	numeric	scalar	true	false
image rotation	task_1	numeric	scalar	true	false

OBJECT VARIABLES

VARIABLE NAME	USED IN TASK	DATA TYPE	DATA FORMAT	IS RECORDED	IS RESETTED

FACTOR VARIABLES

VARIABLE NAME	USED IN TASK	DATA TYPE	DATA FORMAT	IS RECORDED	IS RESETTED

SYSTEM VARIABLES

VARIABLE NAME	USED IN TASK	DATA TYPE	DATA FORMAT	IS RECORDED	IS RESETTED
Subject_Code	Unused	string	scalar	true	false
Subject_Nr	Unused	numeric	scalar	true	false
Subject_Nr_Per_Group	Unused	numeric	scalar	true	false

3.7 Finish the experiment with success

route: /finishExpSession
type: POST
Called when: The experiment is finished successfully
Function on Lavanced Server: Finishes the study and flags the data set as completed (important for balancing).
Payload:

Field name	description	data type	required
expId	Unique identifier for the study	integer	yes
expSessionNr	Unique identifier for the recording session for this participant (counting upwards)	string	yes
end_time	Time when the experiment finished	Unix Timestamp	yes

var_data	Same as the var data in the “/addMetaInfo” route (updated info)	JSON	yes
nextStartTime	Next start time (for longitudinal studies only)	Unix Timestamp	no
nextEndTime	Next end time (for longitudinal studies only)	Unix Timestamp	no
reminderTime	Time until next start time (for longitudinal studies only)	Time string	no
selectedEmail	Email to send participation reminder (for longitudinal studies only)	Email address	no
emailReminder	When to send the reminder (for longitudinal studies only)	string	no

3.8 Finish the experiment with failure

route: /errExpSession
type: POST
Called when: The experiment is aborted with an error
Function on Lavanced Server: Aborts the study and flags the data set as incomplete
Payload:

Field name	**description**	**data type**	**required**
expId	Unique identifier for the study	integer	yes
expSessionNr	Unique identifier for the recording session for this participant (counting upwards)	string	yes
err_msg	The error message	string	yes