

Purdue Farmers Market App API Documentation

Important:

All API requests should be made to a server that is running the software under the path "http://ip:port/api/" (note the /api/). So, for example, if a api endpoint is specified in this document to have a location of "/user/login", that request should be sent to "http://ip:port/api/user/login".

Basic Data IO:

The API requests work via sending and receiving JSON data through HTTP POST requests. All request parameters should be sent under the HTTP request body, and response data will always be sent in the following format (if the page is valid):

```
1  {
2      success: boolean,
3      errorCode: string, // Only will be here if success = false
4      data: { <page-response-data> } // Only will be here if success = true
5  }
```

There are two potential types of errors that can occur and result in an unsuccessful request. Exceptions and Failures. Exceptions are server-side problems that are most likely the result of a bug. Please contact the developer with the code to fix them. Failures are errors that occur because of invalid input parameters. Likely a field is missing or invalid, causing the failure to occur. Only occasionally will failures be the result of a server-side problem.

Whenever an API endpoint has response parameters specified, they will be contained inside the "data" json object (in line 4 above). If it does not specify any response parameters, the "data" json object will be null, and you can get the result of the operation from the "success" boolean.

API Request/Response Example

As an example, if a login request was sent (see /user/login), you would send

```
1  {  
2    email_address: "test@example.com",  
3    password: "secr3t_p3sw0rd",  
4  }
```

And the response from the server would be

```
1  {  
2    success: true,  
3    errorCode: null,  
4    data: {  
5      session_full_code: "1:facc853ea87acdf26852088777f33035"  
6    }  
7  }
```

But, if an error was to occur, the response would be

```
1  {  
2    success: false,  
3    errorCode: "LI-1",  
4    data: null  
5  }
```

This is the basics of the API request/response structure.

Directories and API Endpoints

/user (Directory)

All API endpoints related to user accounts and session will go here.

/user/login

Used for logging in and creating a user session with the given user account.

Request Parameters:

Name	Type	Required	Notes
email_address	string	Yes	The email address of the user you wish to login as
password	string	Yes	The associated password for the user with the specified email

Response Parameters:

Name	Type	Notes
session_full_code	string	The session code used for accessing methods specific to the user you are logged in as. Store this code in order to access other api endpoints that require a session code.

Error Codes:

W-0: Failure - Empty or null request body

W-1: Exception - An unknown error occurred in connecting to the database.

W-2: Exception - An unknown error occurred in the server.

LI-1: Failure - Invalid email or password.

LI-2: Exception - An unknown error occurred in creating a session.

LI-3: Exception - An unknown error occurred in finding a user.

/user/logout

Used to logout from the server. Attempting to use the session_full_code after getting a success from this endpoint will result in an authentication error.

Request Parameters:

Name	Type	Required	Notes
session_full_code	string	Yes	The session code for the session you wish to logout of

Response Parameters:

None. Will return success if successful.

Error Codes:

W-0: Failure - Empty or null request body

W-1: Exception - An unknown error occurred in connecting to the database.

W-2: Exception - An unknown error occurred in the server.

W-3: Failure - session_full_code was empty or null.

W-4: Failure - session_full_code was invalid. Make sure its a valid session code received from /user/login.

W-5: Exception - An unknown error occurred in the server.

W-6: Exception - An unknown error occurred in finding the session.

LO-1: Exception - An unknown error occurred in logging out.

/user/signup

Used to create an account that can be used for logging in.

Request Parameters:

Name	Type	Required	Notes
email_address	string	Yes	
name	string	Yes	The first and last name of the user, will be shown to others.
password	string	Yes	Password used to log in.
user_type	"user" — "vendor"	Yes	The type of the user, can either be normal user (consumer) or a vendor.

Response Parameters: None. Will return success if successful.