D & A
TECHNOLOGIES

Backend Developer Test

Summary

For the D & A Technologies Backend Developer Test, you will be required to develop a PHP API Backend that is designed to be used with a mobile chat application. You will be writing PHP scripts that interact with a MySQL Database, as well as architecting the MySQL Database's structure based on the App's requirements. You will also be required to run and host this code on your own PHP Webserver with a MySQL Database.

Requirements

Imagine that you have been asked to create a RESTful Backend API for a simple mobile chat application. In this chat application, users can create an account, login, and send individual messages to other users in the app. Below you will see a list of required API Endpoints that **you must implement**. Besides these API Endpoints, you are free to structure the PHP scripts and MySQL database any way you like. All API endpoints will json_encode the response data that they send back to the requester.

The only other restriction is that **you are not** allowed to use any PHP Frameworks such as CakePHP, Zend, CodeIgnigter, etc… We recommend using mysqli or PDO to connect to your MySQL Database. It is also recommend that you use some form of OOP to keep your PHP code organized. An example login.php script is hosted here, feel free to send POST requests to this script to see what kind of functionality we are expecting for the various API Endpoints.

Error Response

Some API Endpoints will require you to return an error message or response back to the requester. The only requirements for an error response is that is must have an error_code, error_title, and error_message. You may decide the details of exactly what data goes inside each error response.

```
{
    "error_code": 101
    "error_title": "Login Failure"
    "error_message": "Email or Password was Invalid!"
}
```

D & A
TECHNOLOGIES

API Endpoints

Script Name	Request Type	Description	Example Request Body	Example Response
register.php	POST	Handles registering a brand new user.	**{ “email”:”info@datechnologies.co”, “password”:”Test123”, “first_name”:”John”, “last_name”:”Doe” }**	**{ “user_id”:1, “email”:”info@datechnologies.co”, “first_name”:”John”, “last_name”:”Doe” }**
login.php	POST	Handles a login request from the user. If the login was unsuccessful, you must send an appropriate error response.	**{ “email”:”info@datechnologies.co”, “password”:”Test123” }**	**{ “user_id”:1, “email”:”info@datechnologies.co”, “first_name”:”John”, “last_name”:”Doe” }**

view_messages.php	GET	Returns all messages that these two users have sent to each other in date order.	**{
			"user_id_a": "1",
			"user_id_b": "2"
			}**
				**{
				"messages":[
				{
				"message_id":1,
				"sender_user_id":1,
				"message":"Hey what is up?",
				"epoch":1429220026
				},
				{
				"message_id":2,
				"sender_user_id":2,
				"message":"Not much, how are you doing?",**

				"epoch":1429320028 **}** **]** **}**
send_message.php	POST	Sends a message from one user to another. Returns a success code if the message was sent successfully.	**{** **“sender_user_id”: 1,** **“receiver_user_id”: 2,** **“message” : “Example text”** **}**	**{** **“success_code” : “200”,** **“success_title” : “Message Sent”,** **“success_message”: “Message was sent successfully”** **}**
list_all_users.php	GET	Displays all of the users that have registered to use the app excluding the requester.	**{** **“requester_user_id”: 3** **}**	**{** **"users":[** **{** **"user_id":1,** **"email":"ppeck@datechnologies.co",** **"first_name":"Preston",** **"last_name":"Peck"** **},** **{** **"user_id":2,** **"email":"jgreen@datechnologies.co",** **"first_name":"Jake ",**

				"last_name":"Green" **}** **]** **}**

Deliverables

- All of the PHP scripts and classes that you created for the Test.
- A SQL dump of your MySQL Database in a .sql file so that we can see how you structured your database.
- A READ_ME.txt file that includes the following:
    - Your name.
    - How long it took you to complete the test.
    - A list of general steps you took to complete the test from start to finish.
    - A link to your register.php script (i.e http://dev.datechnologies.co/Tests/scripts/user-signup.php)
    - Notes on possible issues with how the endpoints are currently structured, how they could be improved, any possible security issues with the current implementation.
    - If you were were designing the endpoints yourself, what changes would you make?
- All of these files should be submitted in a single zip file named with the following format: “lastname_firstname_backend_test.zip”