

## INF653 Back End Web Development - Midterm Project Requirements

- 1) You will build a PHP OOP REST API for quotations - both famous quotes and user submissions
- 2) ALL quotes are required to have ALL 3 of the following:
  - a) Quote (the quotation itself)
  - b) Author
  - c) Category
- 3) Create a database named "quotesdb" with 3 tables and these specific column names:
  - a) quotes (id, quote, author\_id, category\_id) - the last two are foreign keys
  - b) authors (id, author)
  - c) categories (id, category)
  - d) id is the primary key in each table
  - e) The id column should also auto-increment
  - f) All columns should be non-null
  - g) 🖱️ Support for this creation process will be documented in Blackboard
- 4) Response requirements:
  - a) All requests should provide a JSON data response.
  - b) All requests for quotes should return the id, quote, author (name), and category (name)
  - c) All requests for authors should return the id and author fields.
  - d) All requests for categories should return the id and category fields.
  - e) Appropriate not found and missing parameter messages as indicated below.
- 5) Your root project URL should follow this pattern ending in /api:  
<https://your-project-name.your-github-username.repl.co/api>

🖱️ Support and documentation will be provided in Blackboard on how to deploy your GitHub repository to your Repl.it account.

The request URLs below should be appended to the root URL.

- 6) Your REST API will provide responses to the following **GET** requests:

Request:	Response:
/quotes/	All quotes are returned
/quotes/?id=4	The specific quote
/quotes/?author_id=10	All quotes from author_id=10
/quotes/?category_id=8	All quotes in category_id=8
/quotes/?author_id=3&category_id=4	All quotes from authorId=3 that are in category_id=4
If no quotes found for routes above	{ message: 'No Quotes Found' }
/authors/	All authors with their id
/authors/?id=5	The specific author with their id

If no authors found for routes above	{ message: 'author_id Not Found' }
/categories/	All categories with their ids (id, category)
/categories/?id=7	The specific category with its id

If no categories found for routes above	{ message: 'category_id Not Found' }
---	--------------------------------------

**NOTE:** In the above examples, the parameter numbers are examples. You could change the authorId=2 or categoryId=5, etc. and the requests should still have the appropriate response.

7) Your REST API will provide responses to the following **POST** requests:

<b>Request:</b>	<b>Response (fields):</b>
/quotes/	created quote (id, quote, author_id, category_id)
<b>Note:</b> To create a quote, the POST submission MUST contain the quote, author_id, and category_id.	
/authors/	created author (id, author)
<b>Note:</b> To create an author, the POST submission MUST contain the author.	
/categories/	created category (id, category)
<b>Note:</b> To create a category, the POST submission MUST contain the category.	
author_id does not exist	{ message: 'author_id Not Found' }
category_id does not exist	{ message: 'category_id Not Found' }
If missing any parameters	{ message: 'Missing Required Parameters' }

8) Your REST API will provide responses to the following **PUT** requests:

<b>Request:</b>	<b>Response (fields):</b>
/quotes/	updated quote (id, quote, author_id, category_id)
<b>Note:</b> To update a quote, the PUT submission MUST contain the id, quote, author_id, and category_id.	
/authors/	updated author (id, author)
<b>Note:</b> To create an author, the PUT submission MUST contain the id and author.	
/categories/	updated category (id, category)
<b>Note:</b> To create a category, the PUT submission MUST contain the id and category.	
author_id does not exist	{ message: 'author_id Not Found' }
category_id does not exist	{ message: 'category_id Not Found' }
If no quotes found to update	{ message: 'No Quotes Found' }
If missing parameters (except id)	{ message: 'Missing Required Parameters' }

9) Your REST API will provide responses to the following **DELETE** requests:

<b>Request:</b>	<b>Response (fields):</b>
-----------------	---------------------------

/api/quotes/	id of deleted quote
/api/authors/	id of deleted author
/api/categories/	id of deleted category
If no quotes found to delete	{ message: 'No Quotes Found' }

**Note:** All delete requests require the id to be submitted.

10) Your project should have a GitHub repository with a README.md file. The README should include your name and a link to your project's home page on Repl. The GitHub repository will be linked to your Repl.it account where the PHP will be deployed. The project should utilize Postgres on Render.com for the database. If you are not familiar with Postgres, it is suggested to become familiar with it in your local dev environment before attempting to host it online with Render.com.

11) You will need to populate your own quotes database. You may choose to populate the database manually or with Postbird (or other management tool) to start out. A good site to find quotes by category (topic) is: <https://www.brainyquote.com/> Minimum 5 categories. Minimum 5 authors. Minimum 25 quotes total for initial data. You will need these minimums to pass the tests for the project.

12) **Submit the following:**

- A link to your GitHub code repository with README.md file included (no updates after the due date accepted)
- A link to your deployed project on Repl
- A one page PDF document discussing what challenges you faced while building your project.

✓ Test your HTTP method requests and responses with Postman: <https://www.postman.com/downloads/>

 **For students who want an extra challenge (not required):**

Allow a "random=true" parameter to be sent via GET request so the response received does not always contain the same quote. The response should contain a random quote that still adheres to the other specified parameters. For example, this will allow users of the API to retrieve a single random quote, a single random quote from Bill Gates (author), or a single random quote about life (category).

**Examples of Extra Challenge Requests and Responses:**

<b>Request:</b>	<b>Response (fields):</b>
/quotes/?random=true	1 random quote
/quotes/?author_id=7&random=true	1 random quote from the specified author
/quotes/?category_id=10&random=true	1 random quote from the specified category
/quotes/?author_id=7&category_id=10&random=true	1 random quote from the specified author & category

## Rubric

Criteria	Expert	Proficient	Developing	Needs Improvement	
API: GET Requests	33 pts Functions as required.	24 pts Meets most requirements	21 pts Meets some requirements	0 pts Does not meet requirements	/33
API: POST Requests	11 pts Functions as required.	8 pts Meets most requirements	5 pts Meets some requirements	0 pts Does not meet requirements	/11
API: PUT Requests	12 pts Functions as required.	8 pts Meets most requirements	5 pts Meets some requirements	0 pts Does not meet requirements	/12
API: DELETE Requests	7 pts Functions as required.	5 pts Meets most requirements	2 pts Meets some requirements	0 pts Does not meet requirements	/7
Public Webpage (index.php)	7 pts Functions as required.	0 pts Does not meet requirements	0 pts Does not meet requirements	0 pts Does not meet requirements	/7
Required Data	10 pts Data as required.	8 pts Most data as required.	7 pts Some data as required.	2.5 pts Very little data.	/10
README.md in GitHub Repository	5 pts README as required.	0 pts No README.	0 pts No README.	0 pts No README.	/10
One Page Discussion Document	10 pts Full Page as required.	4 pts Some Discussion.	0 pts No Discussion.	0 pts No Discussion.	/10
<b>Total</b>					/100

## Overall Score

Expert 90 pts minimum	Proficient 80 pts minimum	Developing 70 pts minimum	Needs Improvement 0 pts minimum
--------------------------	------------------------------	------------------------------	------------------------------------

## Mandatory Requirements:

*You MUST have the following in place to be eligible for any Rubric points above:*

- A GitHub repository for your PHP project
- Your PHP project deployed to Repl
- Your PostgreSQL database deployed to Render.com
- Support & documentation provided for the mandatory requirements.
- Failure to meet these mandatory requirements results in zero points.