Tutor JS Backend

VIP Intelligent Tutoring Systems Spring 2021

The Team

- Project Lead: Lisa Walkosz-Migliacio
- Back End team:
  - Chinmayi Kompella, Colin Cassell, Mike Keohane, Jinwoo Park, Krishan Patel

Motivation

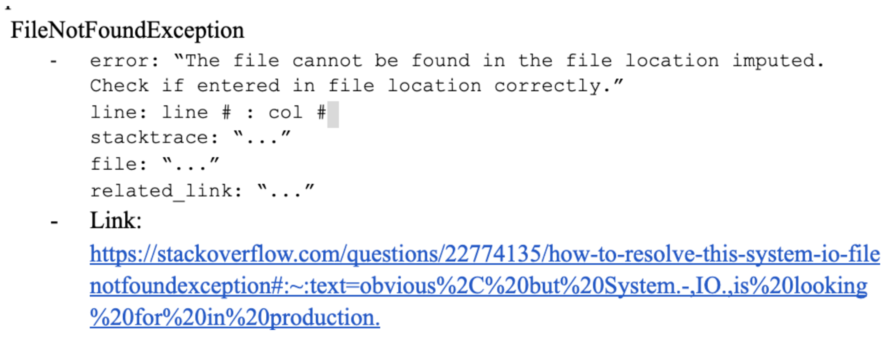
- Build a RESTful API that can communicate with our frontend web application
- Design a database schema that will store the necessary questions, errors, results, and user data for the application
- Connect the React app with the database

Project Outline

- node.js runtime environment
- MongoDB for database, using the mongoose library
- Each Collection is split into 3 parts
  - Model
  - Route
  - Controller
- Axios to make the HTTP requests within the React app

Human Readable Error Messages Research

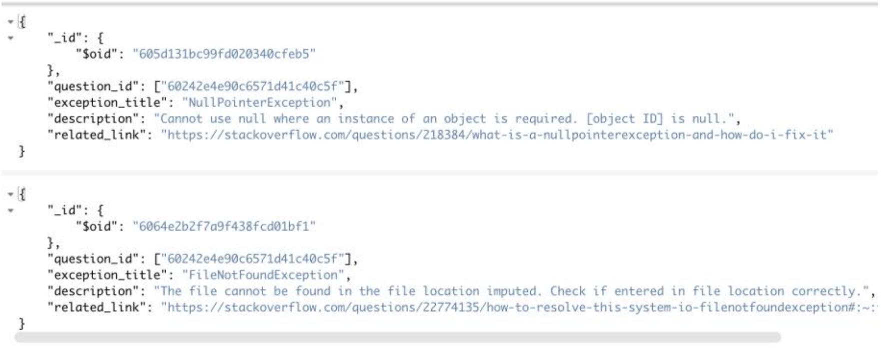
- Goal to create messages that are easily comprehendible by new programmers
  - Many different error messages with vague stacktraces
  - No preexisting code library out there currently
    - Multiple papers created that talked about creating messages
    - Have to create our own error message library
- Messages are within database
  - Make consistent with Ace Editor



FileNotFoundException

```
-   error: "The file cannot be found in the file location imputed.
    Check if entered in file location correctly."
    line: line # : col #
    stacktrace: "..."
    file: "..."
    related_link: "..."
```

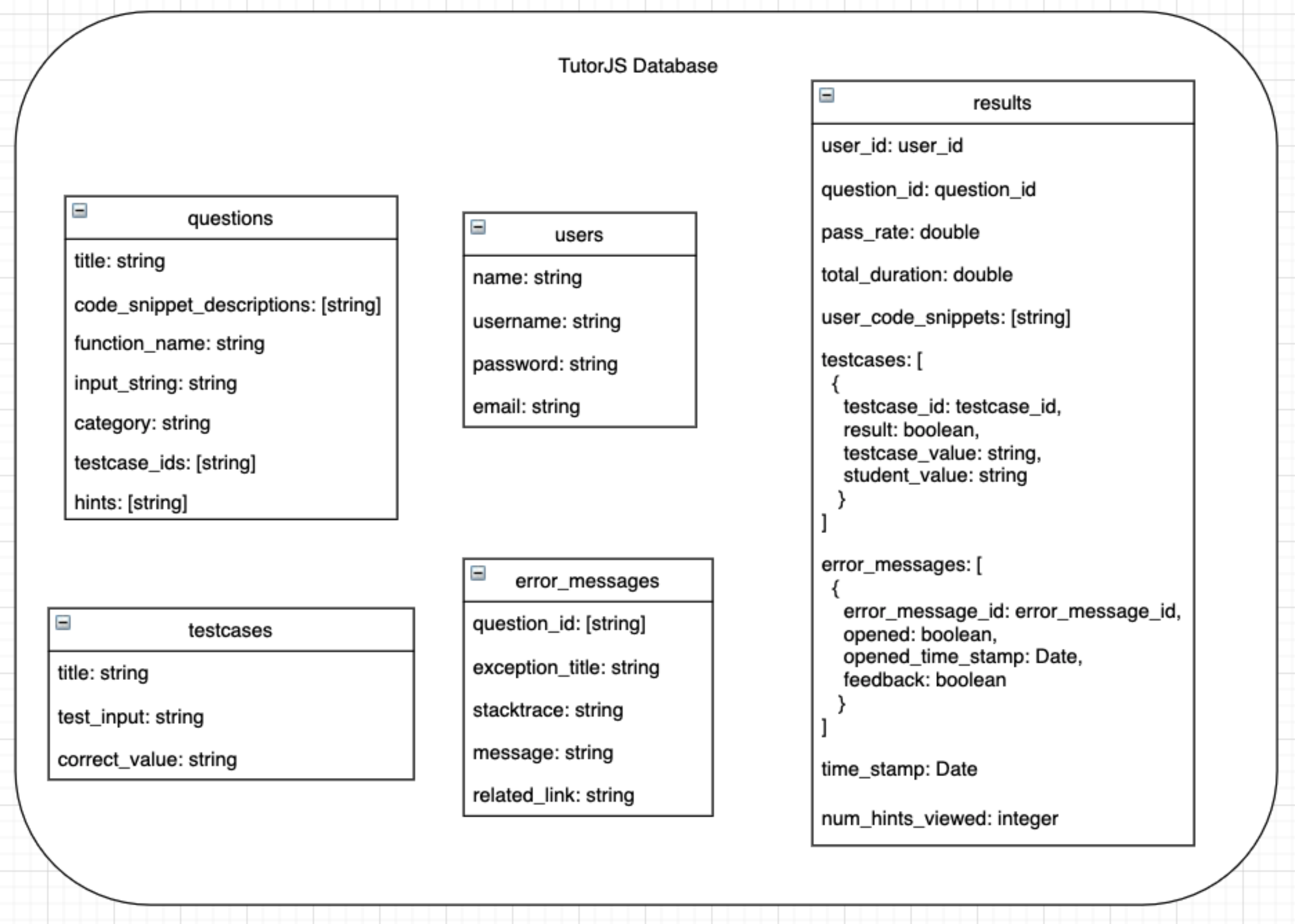
- Link: https://stackoverflow.com/questions/22774135/how-to-resolve-this-system-io-filenotfoundexception#:~:text=obvious%2C%20but%20System.-,IO.,is%20looking%20for%20in%20production.



```
{
    "_id": {
        "$oid": "605d131bc99fd020340cfeb5"
    },
    "question_id": ["60242e4e90c6571d41c40c5f"],
    "exception_title": "NullPointerException",
    "description": "Cannot use null where an instance of an object is required. [object ID] is null.",
    "related_link": "https://stackoverflow.com/questions/218384/what-is-a-nullpointerexception-and-how-do-i-fix-it"
}
{
    "_id": {
        "$oid": "6064e2b2f7a9f438fcd01bf1"
    },
    "question_id": ["60242e4e90c6571d41c40c5f"],
    "exception_title": "FileNotFoundException",
    "description": "The file cannot be found in the file location imputed. Check if entered in file location correctly.",
    "related_link": "https://stackoverflow.com/questions/22774135/how-to-resolve-this-system-io-filenotfoundexception#:~:
}
```

Backend: Database Schema

- We created a database in MongoDB that has 5 collections to hold our data
- The testcases, questions, and error_messages collections hold data that the application uses
- The users and results collections store user data that users generate by interacting with the application



TutorJS Database

```
questions
title: string
code_snippet_descriptions: [string]
function_name: string
input_string: string
category: string
testcase_ids: [string]
hints: [string]
```

```
users
name: string
username: string
password: string
email: string
```

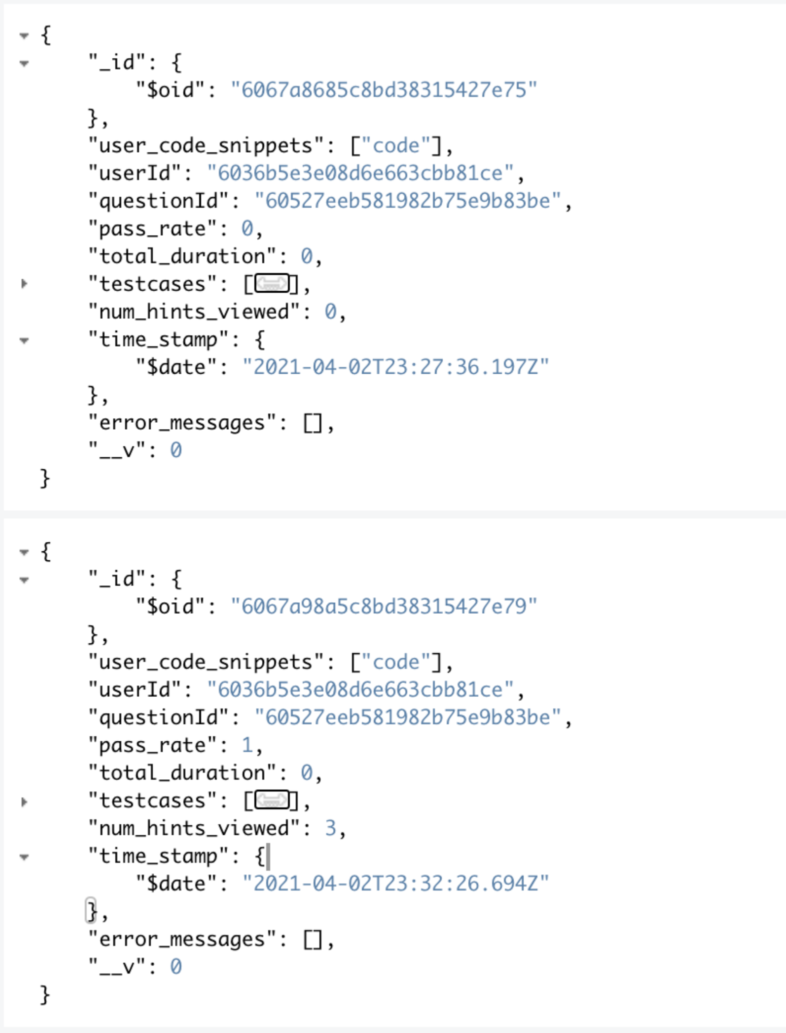
```
results
user_id: user_id
question_id: question_id
pass_rate: double
total_duration: double
user_code_snippets: [string]
testcases: [
 {
   testcase_id: testcase_id,
   result: boolean,
   testcase_value: string,
   student_value: string
 }
]
error_messages: [
 {
   error_message_id: error_message_id,
   opened: boolean,
   opened_time_stamp: Date,
   feedback: boolean
 }
]
time_stamp: Date
num_hints_viewed: integer
```

```
testcases
title: string
test_input: string
correct_value: string
```

```
error_messages
question_id: [string]
exception_title: string
stacktrace: string
message: string
related_link: string
```

Backend: The Results Collection

- These are the user results that we want to store for future analysis
- By storing results like these, we can clearly see how users progress from one attempt to another
- With this method of logging data, we can see which hints and error messages were more or less helpful to students

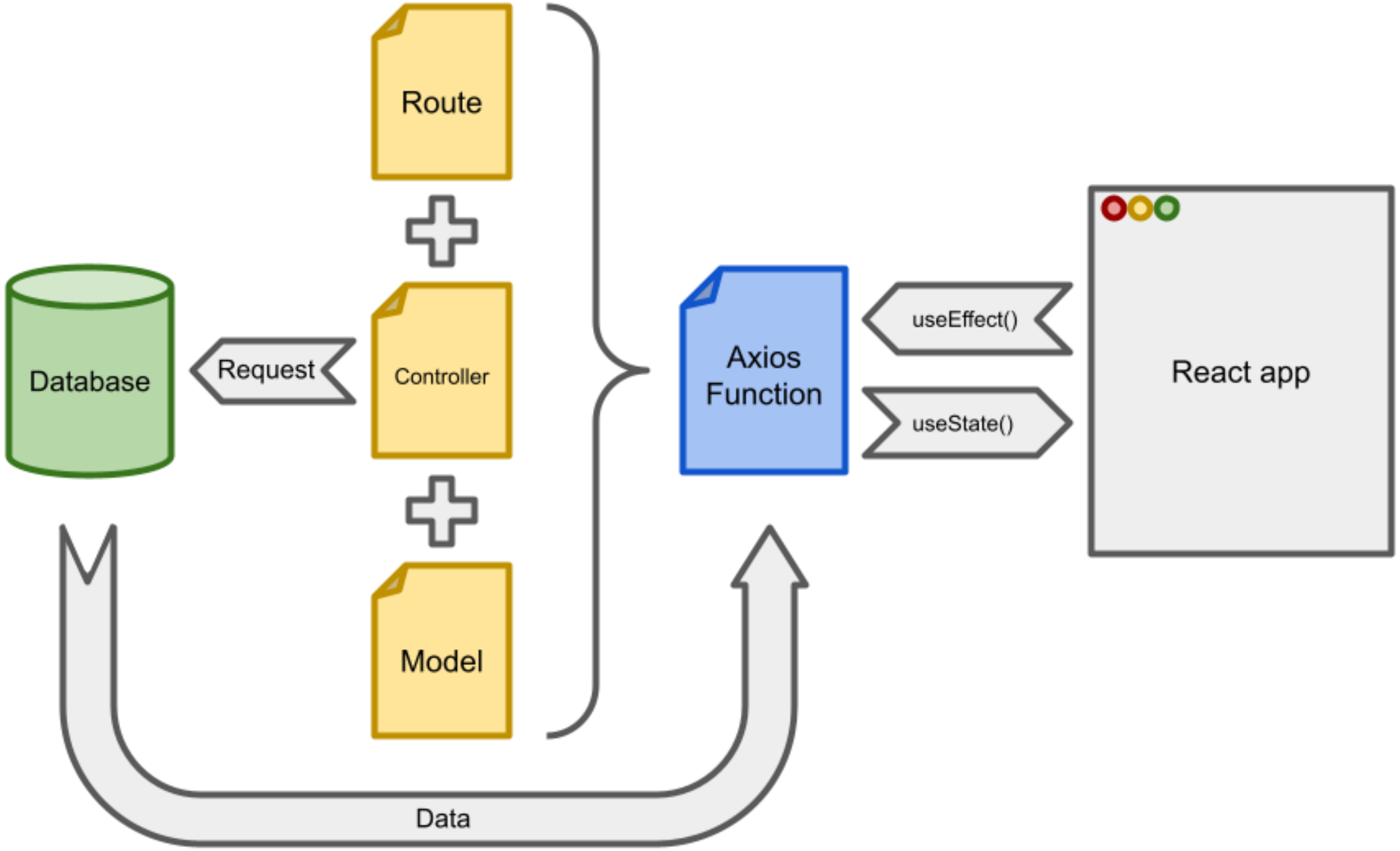


```
{
    "_id": {
        "$oid": "6067a8685c8bd38315427e75"
    },
    "user_code_snippets": ["code"],
    "userId": "6036b5e3e08d6e663cbb81ce",
    "questionId": "60527eeb581982b75e9b83be",
    "pass_rate": 0,
    "total_duration": 0,
    "testcases": [↔],
    "num_hints_viewed": 0,
    "time_stamp": {
        "$date": "2021-04-02T23:27:36.197Z"
    },
    "error_messages": [],
    "__v": 0
}
```

```
{
    "_id": {
        "$oid": "6067a98a5c8bd38315427e79"
    },
    "user_code_snippets": ["code"],
    "userId": "6036b5e3e08d6e663cbb81ce",
    "questionId": "60527eeb581982b75e9b83be",
    "pass_rate": 1,
    "total_duration": 0,
    "testcases": [↔],
    "num_hints_viewed": 3,
    "time_stamp": {
        "$date": "2021-04-02T23:32:26.694Z"
    },
    "error_messages": [],
    "__v": 0
}
```

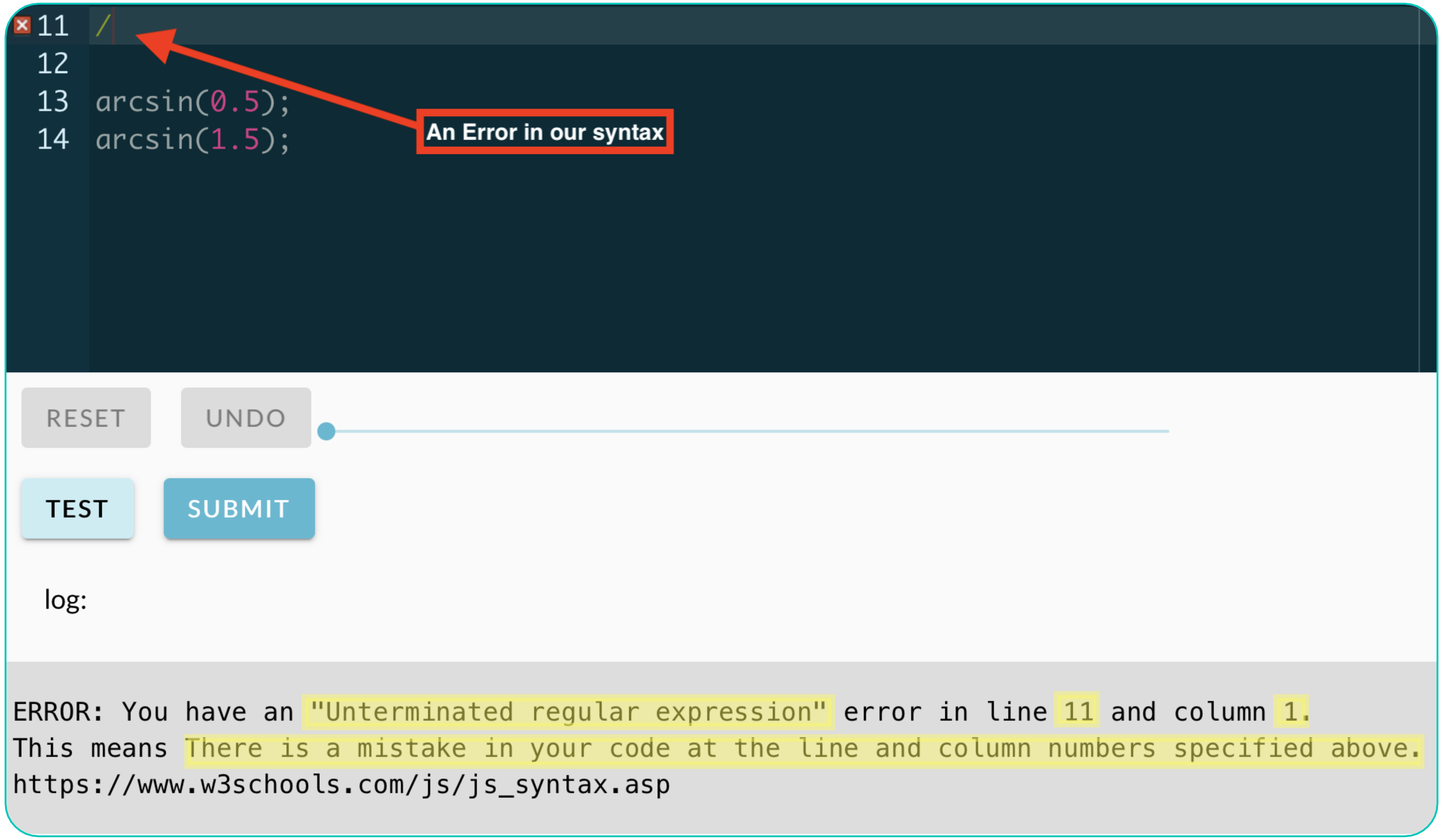
Connecting the backend to React

- React app notices a change in state
- useEffect() calls axios function
- Using the defined route and controller, http request looks for appropriate model in database
- Database sends data back as JSON
- useState() updates React component with retrieved data



Error Message Parsing:

How do we get here?

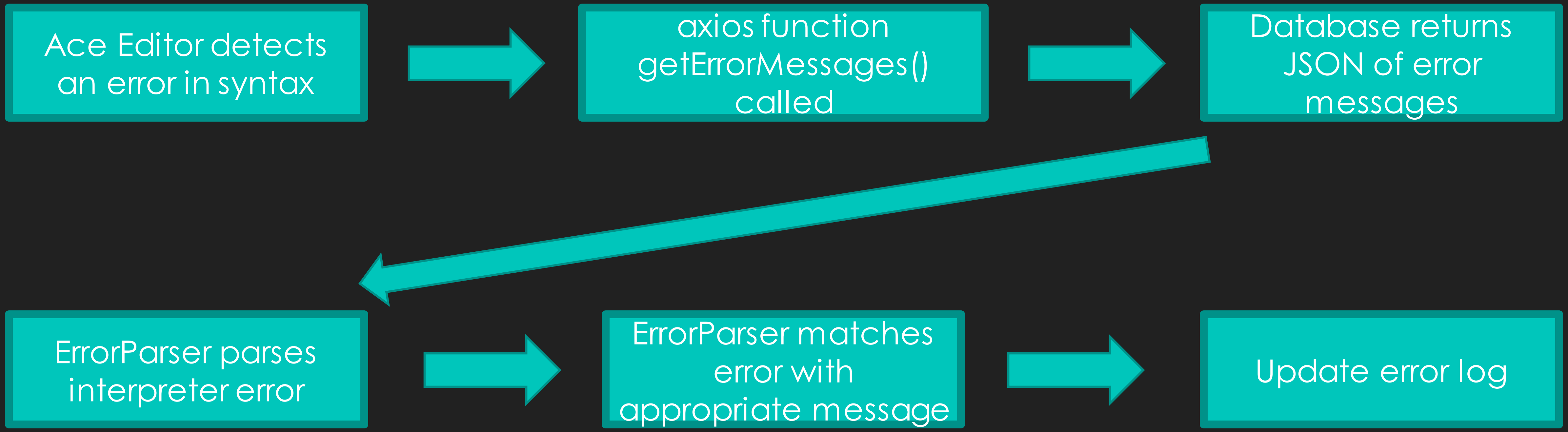


```
11  /
12
13  arcsin(0.5);
14  arcsin(1.5);
```

An Error in our syntax

```
ERROR: You have an "Unterminated regular expression" error in line 11 and column 1.
This means There is a mistake in your code at the line and column numbers specified above.
https://www.w3schools.com/js/js_syntax.asp
```

Timeline of events



Ace Editor detects an error in syntax

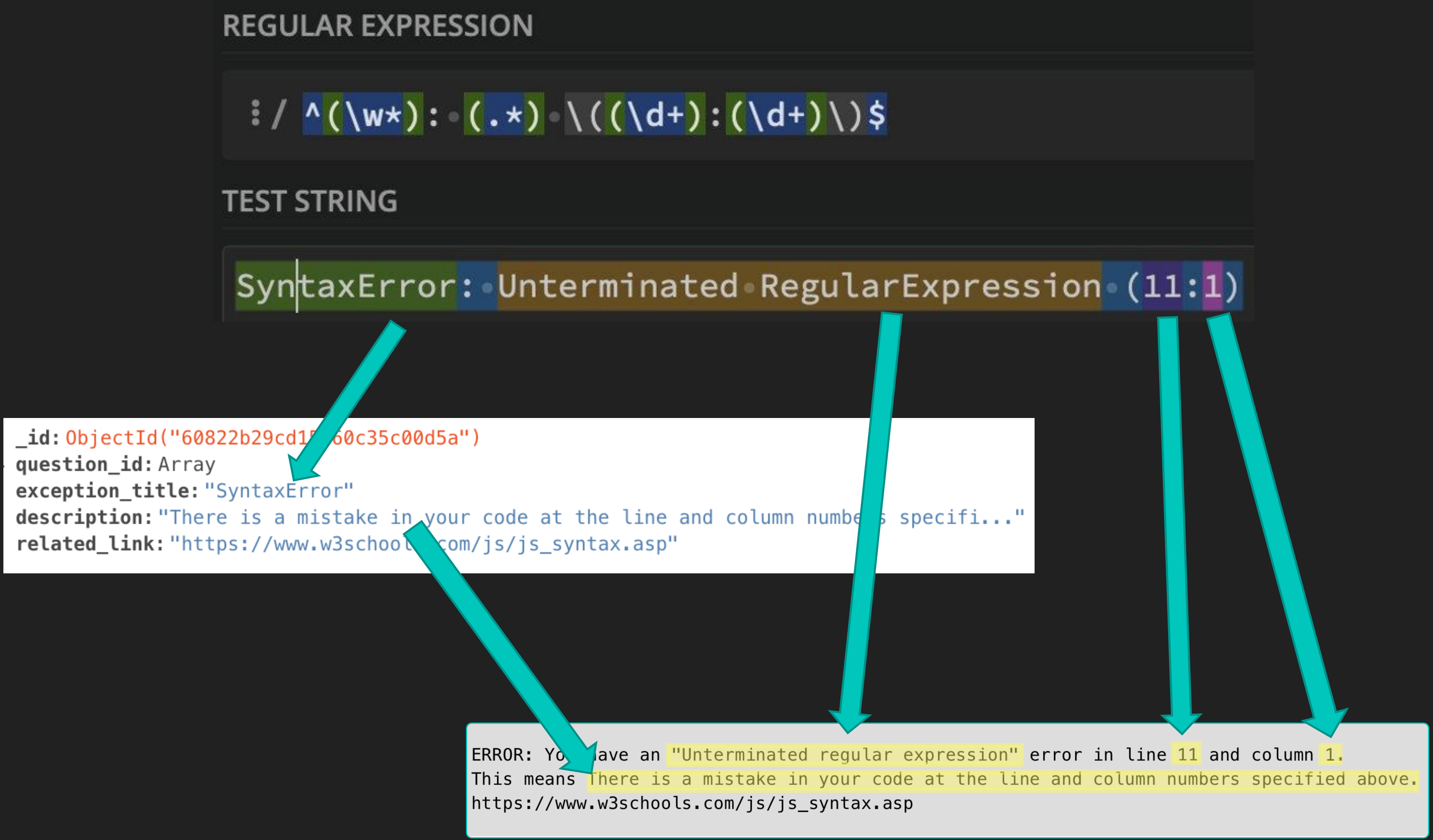
axios function getErrorMessages() called

Database returns JSON of error messages

ErrorParser parses interpreter error

ErrorParser matches error with appropriate message

Update error log



REGULAR EXPRESSION

```
/ ^(\w*): (.*) \((\d+):(\d+)\)$
```

TEST STRING

```
SyntaxError: Unterminated RegularExpression (11:1)
```

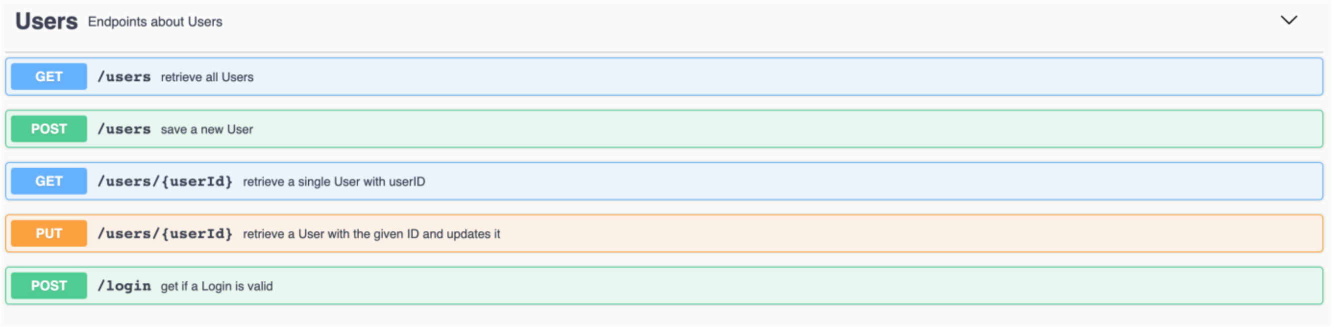
```
_id: ObjectId("60822b29cd1[illegible]60c35c00d5a")
question_id: Array
exception_title: "SyntaxError"
description: "There is a mistake in your code at the line and column numbers specifi..."
related_link: "https://www.w3schools.com/js/js_syntax.asp"
```

```
ERROR: You have an "Unterminated regular expression" error in line 11 and column 1.
This means There is a mistake in your code at the line and column numbers specified above.
https://www.w3schools.com/js/js_syntax.asp
```

Documenting the backend

- Used Swagger to host a clean visualization of endpoints
- Lists all possible API functionalities, how to call each endpoint, and what to expect for return values
- Allows you to directly call the API endpoints from the page
- Makes it easy for future developers to fully utilize our API

Visualize All Endpoints



Users Endpoints about Users

GET `/users` retrieve all Users

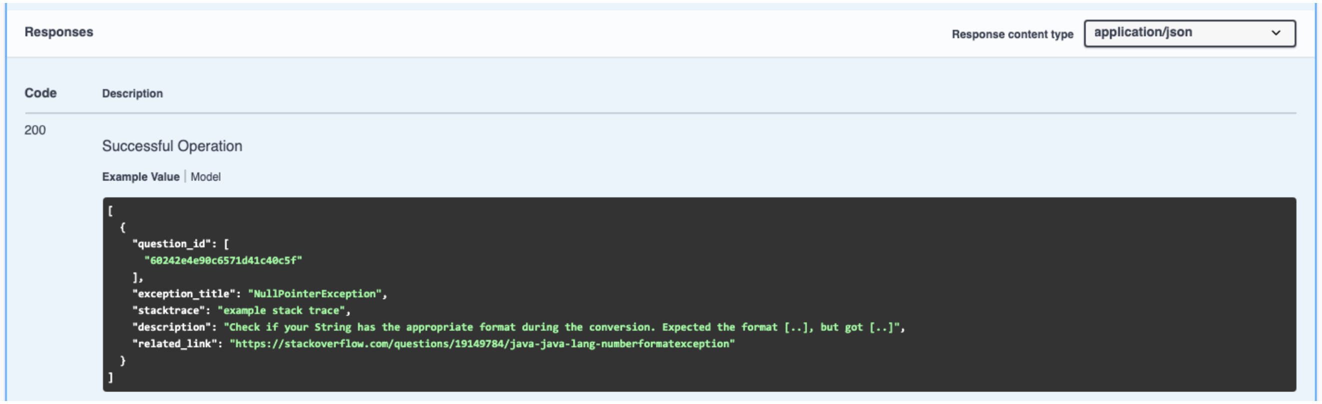
POST `/users` save a new User

GET `/users/{userId}` retrieve a single User with userID

PUT `/users/{userId}` retrieve a User with the given ID and updates it

POST `/login` get if a Login is valid

Viewing Possible Return Values



Responses

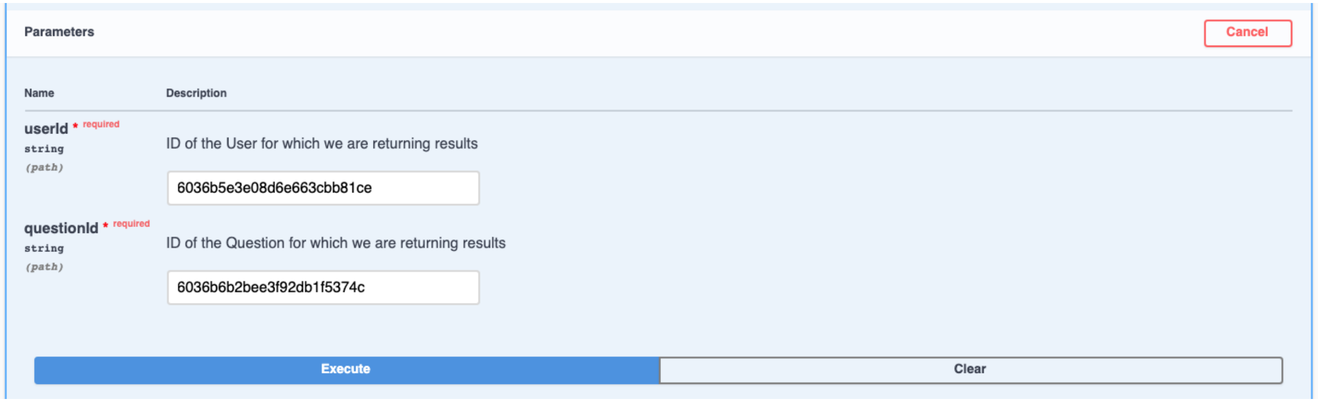
Response content type: application/json

Code	Description
200	Successful Operation

Example Value | Model

```
[
  {
    "question_id": [
      "60242e4e90c6571d41c40c5f"
    ],
    "exception_title": "NullPointerException",
    "stacktrace": "example stack trace",
    "description": "Check if your String has the appropriate format during the conversion. Expected the format [..], but got [..]",
    "related_link": "https://stackoverflow.com/questions/19149784/java-java-lang-numberformatexception"
  }
]
```

Directly Query API



Parameters

Name	Description
userId * required string *(path)*	ID of the User for which we are returning results 6036b5e3e08d6e663cbb81ce
questionId * required string *(path)*	ID of the Question for which we are returning results 6036b6b2bee3f92db1f5374c

Future Features Backend

- Implement better human readable error messages within the IDE
  - Make links more customizable and more relevant to the particular error message
- Backend analytics to make system more "intelligent"
  - Collect how long student has been on question, how many attempts, etc.
  - What are the most commonly missed answers? (integration with chatbot team?)
- Integrate DLTI functions into challenges given in the interface
  - Find ways to break down the DLTI functions into different "problems" to allow student to really understand the functions
  - Output the code into visualizations like the DLTI graphs

DEMO