**List of Pages**

* Homepage
* Answer Question/Lesson view
* Instructor:
  + Manage class
    - Students
      * Add and kick
      * Private or public (students can join using the class code)
    - Questions
      * List of questions (adding/altering)
      * View results
      * Export results
      * Link to submitty?
* Student
  + View classes
    - View questions and scores
  + Join class (through id?)
* **New Less Messy List of Routes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Route** | **Request Body** | **Response Body** |
| **Groups (classes)** | | | |
| GET | /api/groups/<groupID> |  | [  {  group...  }  ] |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

* **List of Routers/Routes (Frontend look here) (red is not working, green is working, yellow is some other issue)**
  + **Groups (classes)**
    - /api/groups
      * GET - gets list of all groups (array)
        + **Testing ONLY**
    - /api/groups/<groupID>
      * GET - gets group metadata (JSON object)
    - /api/groups/<groupID>/polls
      * GET - gets list of polls for the group (array of poll IDs) { [ pollid, pollid ] }
    - /api/groups/new
      * POST - JSON {
        + Name: string
    - /api/groups/<groupID>/edit
      * POST - JSON {
        + Action: String (Add, Remove)
        + Name: String groupname, or
        + InstructorID: ID, or
        + PollID: ID, or
        + UserID: ID
    - /api/groups
      * GET - Prints out groups collection for now
  + **Polls (series of questions)** 
    - /api/polls
      * GET - gets list of all polls (array)
        + **Testing ONLY**
    - /api/polls/<pollID>
      * GET - gets the specified poll object (including poll answers)
    - /api/polls/<pollID>/submit/
      * POST - submit answer JSON
        + UserID: <userID>
        + Answers: Array

Objects

QuestionID: ID

Answer:

String

Int

Boolean

Etc?

* + - /api/polls/new
      * POST - JSON {
        + Name: string
    - /api/polls/<pollID>/edit
      * POST - JSON {
        + Action: String (Add, Remove)
        + Question: Obj {

QuestionText: String

AnswerChoices: Array[String]

CorrectAnswers: Array[String]

* + **Users**
    - /api/users
      * GET - gets list of all users (array)
        + **Testing ONLY**
    - /api/users/login
      * POST - login/create session
      * GET - Could be used to tell if user is successfully logged in
    - /api/users/register
      * POST - JSON {

FirstName: string,

LastName: string,

Username: string,

Email: string,

Password: string

RCS: boolean (Optional, default false)

}

* + - /api/users/<userID>
      * GET - get user metadata
    - /api/users/<userID>/edit
      * POST - change account details (Name, RCS, email) JSON Object
        + Action: String (Add (also used to update), Remove)
        + Send only one of the following:

FirstName: string

LastName: string

Username: string

Email: string

Password: string

RCS: boolean

Groups: GroupID

* + - /api/users/<userID>/groups
      * GET - get user’s groups

**What are the different types of questions?**

Multiple Choice

True/False

Open Text

Checkbox (Single or multiple)

**How are questions presented? ()**

Is the question visible on the client?

**What data needs to be stored?**

*Naming scheme: PascalCase for fields (APPLIES TO JSON TOO), lowercased\_underscoring for collection names*

*Keyname: type of data*

*What’s actually in it*

Collections and Fields:

* users
  + FirstName: String
  + LastName: String
  + Username: String
  + Email: String
  + Password: (hashed and salted) String
  + Groups: Array
    - GroupIDs
  + RCS: Boolean, default false unless RCS account, then:
    - omit password, use RCSID as username, and set RCS = true
* groups
  + Name: String
  + Instructors: Array of InstructorIDs
  + Polls: Array
    - PollIDs
  + Users: Array
    - UserIDs
* polls
  + Name: String
  + Questions: Array (ordered)
    - Documents//fields???
      * ID (needs to be manually generated)
      * QuestionText: String
      * AnswerChoices: Array
        + Strings
        + Ints
        + Booleans
        + Etc?
      * CorrectAnswers: Array (can be omitted for no grading)
        + Contains subset of AnswerChoices
* poll\_answers
  + \_id
  + PollID: ID
  + UserID: ID
  + Answers: Array
    - Documents
      * QuestionID: ID
      * Answer:
        + String
        + Int
        + Boolean
        + Etc?
      * Correct: Boolean