

Milestone 1
Milestone 2 4/9
Milestone 3 4/23

Endpoint	Method	Request Params POST parameters are JSON (application/json)	Description All return types are JSON
/adduser	POST	<ul style="list-style-type: none"> - username: string - email: string - password: string 	<ul style="list-style-type: none"> - Register new user account - Username and email must be unique - Should send email with verification key - Returns: <ul style="list-style-type: none"> - status: “OK” or “error” - error: error message (if error)
/login	POST	<ul style="list-style-type: none"> - username - password 	<ul style="list-style-type: none"> - Login to account - Start session - Returns: <ul style="list-style-type: none"> - status: “OK” or “error” - error: error message (if error)
/logout	POST		<ul style="list-style-type: none"> - Logout of account - Returns: <ul style="list-style-type: none"> - status: “OK” or “error” - error: error message (if error)
/verify	POST	<ul style="list-style-type: none"> - email - key: verification key 	<ul style="list-style-type: none"> - Verifies account - Account cannot be used until account is verified - Email should include the text: <i>validation key: <key_goes_here></i> (including < and > characters) - Returns: <ul style="list-style-type: none"> - status: “OK” or “error” - error: error message (if error)
			<ul style="list-style-type: none"> - Add a backdoor key “abracadabra” for validation

/questions/add	POST	<ul style="list-style-type: none"> - title: title of the question - body: body of the question - tags: array of tags (strings) 	<ul style="list-style-type: none"> - Post a new question - Only allowed if logged in - Returns <ul style="list-style-type: none"> - status: “OK” or “error” - id: unique question ID string (if OK) - error: error message (if error)
		<ul style="list-style-type: none"> - media: array of media IDs <ul style="list-style-type: none"> - Optional 	
/questions/{id}	GET		<ul style="list-style-type: none"> - Get contents of a single question, with {id}, and increments view_count (returns updated count in response) - Views are unique by authenticated users, and for unauthenticated users, unique by IP address. - Returns <ul style="list-style-type: none"> - status: “OK” or “error” - question: { <ul style="list-style-type: none"> - id: string - user: { <ul style="list-style-type: none"> - username: string - reputation: int - } - title: string - body: string - score: int - view_count: int - answer_count: int - timestamp: timestamp, represented as Unix time - media: array of associated media IDs - tags: array of tags - accepted_answer_id: id of accepted answer, if there exists one. Null otherwise - } - error: error message (if error)
/questions/{id}	DELETE		<ul style="list-style-type: none"> - Delete question with {id} - Should only succeed deleting if logged in user is original asker. - Result in HTTP status code <ul style="list-style-type: none"> - 200 OK on Success, 4xx on Failure
			<ul style="list-style-type: none"> - Delete associated media file(s), and answers
/questions/{id}/answers/add	POST	<ul style="list-style-type: none"> - body: string 	<ul style="list-style-type: none"> - Adds an answer to question with specified id - Only allowed if logged in - Returns <ul style="list-style-type: none"> - status: “OK” or “error” - id: unique answer ID string (if OK)

			<ul style="list-style-type: none"> - error: error message (if error)
		<ul style="list-style-type: none"> - media: array of media IDs <ul style="list-style-type: none"> - Optional 	
/questions/{id}/answers	GET		<ul style="list-style-type: none"> - Gets all answers to question with specified id - Returns: <ul style="list-style-type: none"> - Array of answers: “answers”: [<ul style="list-style-type: none"> - { <ul style="list-style-type: none"> - id: string - user: username of poster - body: string - score: int - is_accepted: boolean - timestamp: timestamp, represented as Unix time - media: array of associated media IDs - }, ... -] - status: “OK” or “error” - error: error message (if error)
/search	POST	<ul style="list-style-type: none"> - timestamp: search questions from this time and earlier <ul style="list-style-type: none"> - Represented as Unix time in seconds - Integer, optional - Default: Current time - limit: number of questions to return <ul style="list-style-type: none"> - Integer, optional - Default: 25 - Max: 100 	<ul style="list-style-type: none"> - Gets a list of the latest {limit} number of questions prior to (and including) the provided {timestamp} - Returns <ul style="list-style-type: none"> - status: “OK” or “error” - questions: Array of question objects (see /question/{id}) - error: error message (if error)
		<ul style="list-style-type: none"> - q: search query, <ul style="list-style-type: none"> - String, optional - Should support spaces: <i>For example:</i> “hello world” should return questions with titles OR bodies which contain the phrase “hello world”. 	
		<ul style="list-style-type: none"> - sort_by: Order returned questions by “timestamp” or by “score” <ul style="list-style-type: none"> - String, optional - Default: score - tags: Return only questions which have all of these tags (AND). <ul style="list-style-type: none"> - Array of strings, optional - has_media: Return questions with media only 	

		<ul style="list-style-type: none"> - Boolean - if true, exclude all questions that do not have an associated media, optional - Default: false - accepted: only return questions with accepted answers <ul style="list-style-type: none"> - boolean, optional - Default: false 	
/user/{username}	GET		<ul style="list-style-type: none"> - Gets user profile information for user with {username} - Returns <ul style="list-style-type: none"> - status: "OK" or "error" - user: { <ul style="list-style-type: none"> - email: string - reputation: int - }
/user/{username}/questions	GET		<ul style="list-style-type: none"> - Gets questions posted by user with {username} - Returns <ul style="list-style-type: none"> - status: "OK" or "error" - questions: array of question ids posted by this user
/user/{username}/answers	GET		<ul style="list-style-type: none"> - Gets answers posted by user with {username} - Returns <ul style="list-style-type: none"> - status: "OK" or "error" - answers: array of answer ids posted by this user
/questions/{id}/upvote	POST	<ul style="list-style-type: none"> - upvote: <ul style="list-style-type: none"> - Boolean - Default: true 	<ul style="list-style-type: none"> - Upvotes or downvotes the question (in/decrements) score - Upvoting an already upvoted question (by the same user) should undo the upvote, and vice versa. - Increments/decrements the reputation of the asker. Reputation must always be >=1. - - Returns <ul style="list-style-type: none"> - status: "OK" or "error"
/answers/{id}/upvote	POST	<ul style="list-style-type: none"> - upvote: <ul style="list-style-type: none"> - Boolean - Default: true 	<ul style="list-style-type: none"> - Upvotes or downvotes the answer (in/decrements) score - Upvoting an already upvoted answer (by the same user) should undo the upvote, and vice versa. - Increments/decrements the reputation of the answerer. Reputation must always be >= 1. - - Returns <ul style="list-style-type: none"> - status: "OK" or "error"
/answers/{id}/accept	POST		<ul style="list-style-type: none"> - Accepts an answer - Should only succeed if logged in user is original asker of associated

			<div>question.</div> <div><div>- Returns<ul style="list-style-type: none">- status: “OK” or “error”</div></div>
/addmedia	POST	<div>Type is multipart/form-data</div> <div><div>- content: binary content of file being uploaded</div></div>	<div><div>- Adds a media file (photo or video)</div><div>- Only allowed if logged in</div><div>- Returns<ul style="list-style-type: none">- status: “OK” or “error”- id: ID of uploaded media- error: error message (if error)</div></div>
/media/{id}	GET		<div><div>- Gets media file by {id}</div><div>- Returns<ul style="list-style-type: none">- media file (image or video)- HTTP status code<ul style="list-style-type: none">- 200 on Success, 4xx on Failure</div></div>