

## **Goal-e: Service Layer Design**

Maryville University

Fall 2023

## Service Endpoints

Below are the endpoints I plan on implementing for my project's web server.

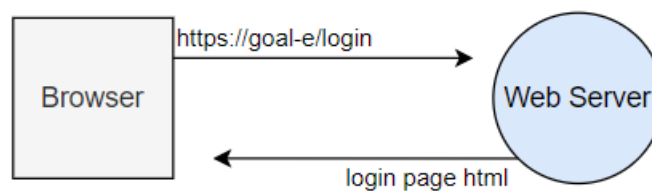
I designed these endpoints based on building my project with the Django framework.

Endpoint	Page
GET /login	2
GET /signup	2
POST /login	3
POST /signup	4
GET /	5
Get /pastGoals	5
GET /calendar/{month}/{year}	6
GET /dailyGoals/{month}/{day}/{year}	7
GET /goals/{goalId}	8
GET /newGoal	9
PUT /goals/{goalId}	9
POST goals/{goalId}/complete	11
POST /newGoal	12
GET /*invalid*	13
GET /accountSettings (stretch goal)	14
PUT /accountSettings (stretch goal)	14

**GET /login**

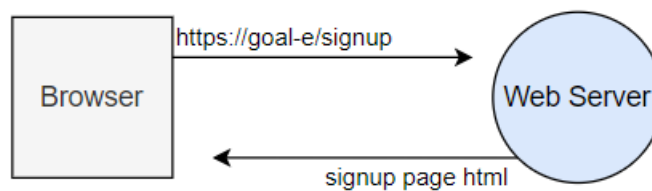
Returns the html for the login page.

Request	Response
Browser: https://goal-e/login	200 Login page html
Browser: https://goal-e/login If already logged-in	302 Redirect to /

**GET /signup**

Returns the html for the sign-up page.

Request	Response
Browser: https://goal-e/signup	200 Signup page html
Browser: https://goal-e/signup If already logged-in	302 Redirect to /



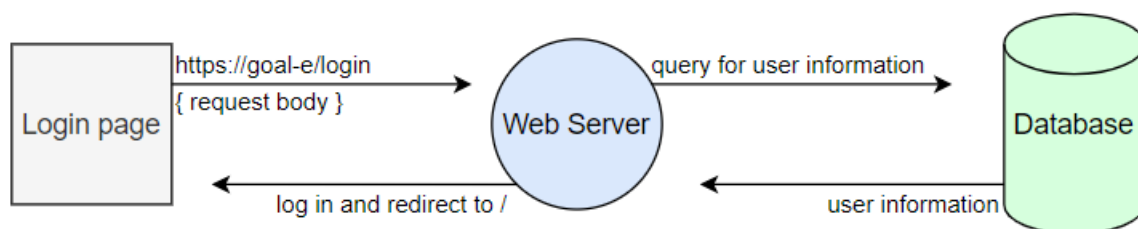
**POST /login**

Attempts to log the user in and, if successful, redirects to /.

Parameters	Description
username	Username string
password	Password string

Request	Response
To: https://goal-e/login Body: <pre>{   "username": "valid_username",   "password": "valid_password" }</pre>	302 Redirect to /
To: https://goal-e/login Body: <pre>{   "username": "invalid_username",   "password": "invalid_password" }</pre>	401 Body: <pre>{   "issues": {     "username": ["issue 1", "issue 2"],     "password": ["issue 1", "issue 2"]   } }</pre>
Browser: https://goal-e/login If already logged-in	302 Redirect to /

\*JSON server response can be used by user interface to display error messages without reloading



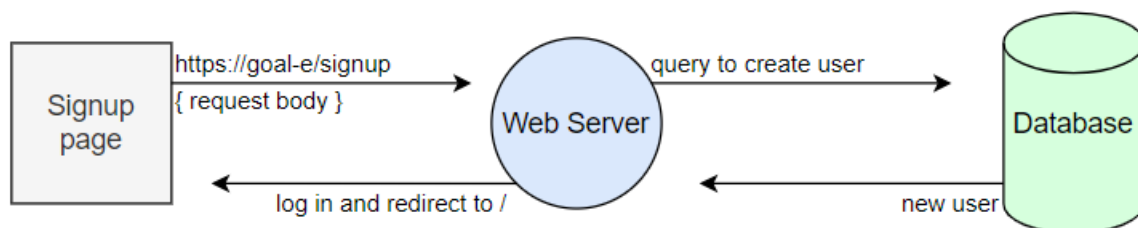
**POST /signup**

Attempts to create a new user in and, if successful, logs the new user in and redirects to /.

Parameters	Description
username	Username string
password	Password string
confirmPassword	Password string

Request	Response
To: https://goal-e/signup Body: <pre>{   "username": "valid_username",   "password": "valid_password",   "confirmPassword": "valid_password" }</pre>	302 Redirect to /
To: https://goal-e/signup Body: <pre>{   "username": "invalid_username",   "password": "invalid_password",   "confirmPassword": "invalid_password" }</pre>	400 Body: <pre>{   "issues": {     "username": ["issue 1", "issue 2"],     "password": ["issue 1", "issue 2"]   } }</pre>
Browser: https://goal-e/signup If already logged-in	302 Redirect to /

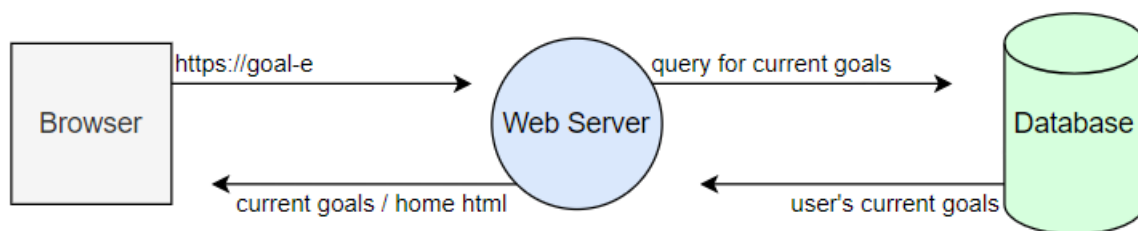
\*JSON server response can be used by user interface to display error messages without reloading



**GET /**

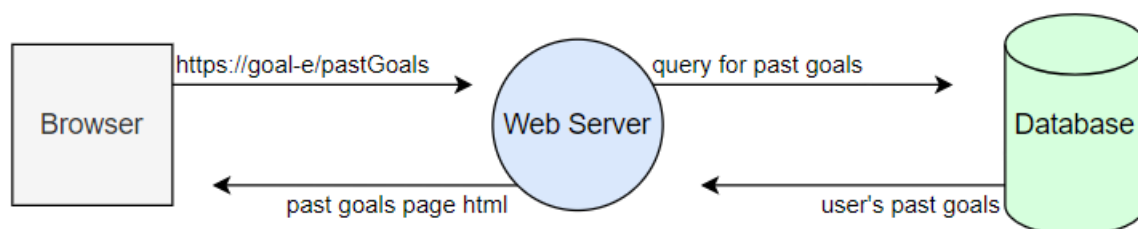
Retrieves the current goals of the logged-in user and returns the html for the home / current goals page rendered with those goals.

Request	Response
Browser: https://goal-e If logged-in	200 Home page / current goals html
Browser: https://goal-e If NOT logged-in	302 Redirect to /login

**GET /pastGoals**

Retrieves the previously completed goals of the logged-in user and returns the html for the past goals page rendered with those goals.

Request	Response
Browser: https://goal-e/pastGoals If logged-in	200 Past goals html
Browser: https://goal-e/pastGoals If NOT logged-in	302 Redirect to /login



**GET /calendar/{month}/{year}**

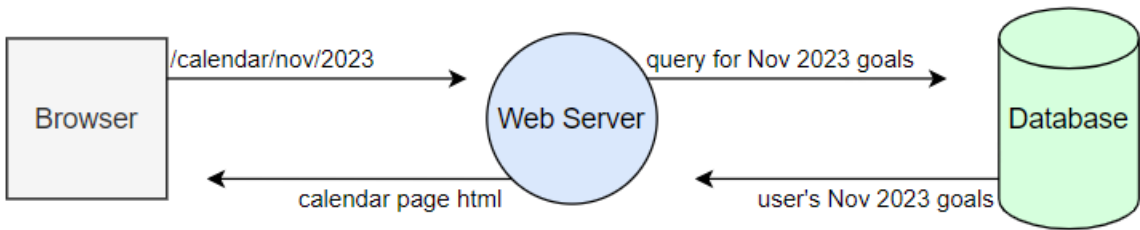
Retrieves the goals of the logged-in user that have a deadline in the specified month. Returns the html for the calendar page based on the specified month rendered with the retrieved goals.

Parameters	Description
month	3 letter month designation (i.e., ‘nov’, ‘dec’)
year	4-digit year designation (i.e., 2023, 2024)

If no month and year parameters are specified, or if invalid parameters are specified, default values of the current month and year should be used.

If a valid month parameter is specified but no year parameter is specified, the current year should be used by default.

Request	Response
Browser: https://goal-e/calendar/nov/2023 If logged-in	200 Calendar page html for November 2023
Browser: https://goal-e/calendar/nov/2023 If NOT logged-in	302 Redirect to /login

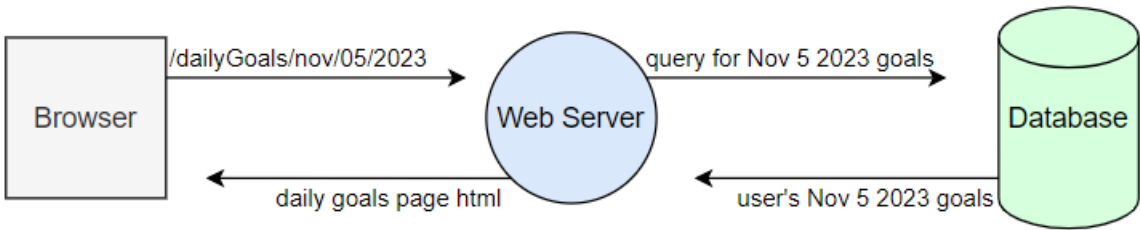


**GET /dailyGoals/{month}/{day}/{year}**

Retrieves the goals of the logged-in user that have a deadline on the specified date. Returns the html for the daily goals page based on the specified date rendered with the retrieved goals.

Parameters	Description
month	3 letter month designation (i.e., ‘nov’, ‘dec’)
year	4-digit year designation (i.e., 2023, 2024)
day	2-digit day designation (i.e., 01, 15)

Request	Response
Browser:  https://goal-e/dailyGoals/nov/05/2023  If logged-in	200  Daily goals html for November 5, 2023
Browser:  https://goal-e/dailyGoals/nov/05/2023  If NOT logged-in	302  Redirect to /login



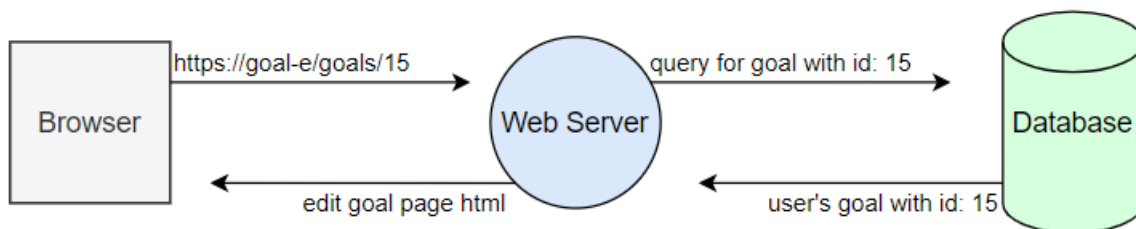


**GET /goals/{goalId}**

Retrieves the goal of the logged-in user corresponding to the specified id. Returns the html for the edit goal page rendered with the retrieved goal.

Parameters	Description
goalId	ID of goal to return

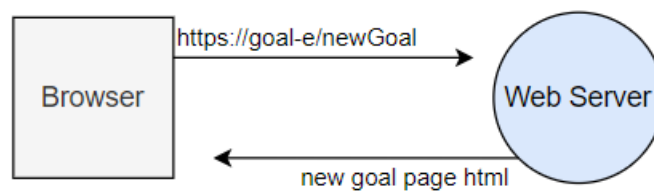
Request	Response
Browser: https://goal-e/goals/15 If logged-in	200 Edit goal html for goal with id of 15
Browser: https://goal-e/goals/15 If logged-in AND invalid goal	404 Goal not found html
Browser: https://goal-e/goals/15 If NOT logged-in	302 Redirect to /login



**GET /newGoal**

Returns the html for the new goal page.

Request	Response
Browser: https://goal-e/newGoal If logged-in	200 New goal html
Browser: https://goal-e/newGoal If NOT logged-in	302 Redirect to /login

**PUT /goals/{goalId}**

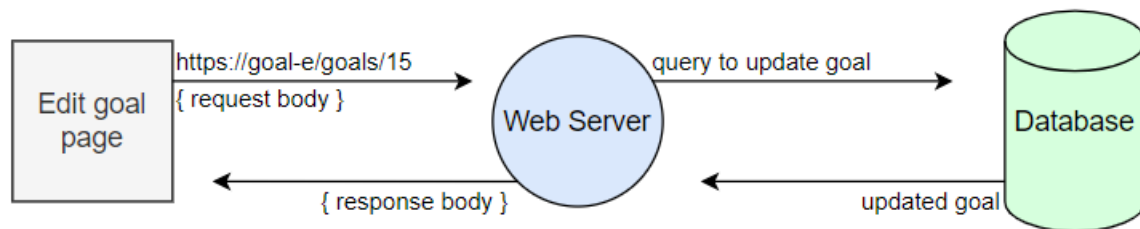
Updates the goal with the specified id.

Parameters	Description
title	Goal title
description	Goal description
deadline	Date the goal should be completed by
progress	Number between 0 and 100 indicating the progress towards goal completion
priority	Number between 1 and 3 representing the goal's priority

Request	Response
To: https://goal-e/goals/15 Body: <pre>{     "title": "Walk 1000 steps",     "description": "description...",     "deadline": "11/05/2023",     "progress": 25,     "priority": 2   }</pre>	200  Body: <pre>{     "message": "successfully updated goal!"   }</pre>

To: https://goal-e/goals/15 Body: <pre>{   "title": "invalid title",   "description": "invalid description",   "deadline": "invalid deadline",   "progress": "invalid progress",   "priority": "invalid priority" }</pre>	400 Body: <pre>{   "title": ["issue 1"],   "description": ["issue 1"],   "deadline": ["issue 1"],   "progress": ["issue 1"],   "priority": ["issue 1"] }</pre>
To: https://goal-e/goals/15 If invalid goal id	404 Body: <pre>{   "error": "Goal with id: 15 not found" }</pre>
To: https://goal-e/goals/15 If NOT logged-in	401 Body: <pre>{   "error": "Not logged in" }</pre>

\*JSON server response can be used by user interface to update information without reloading

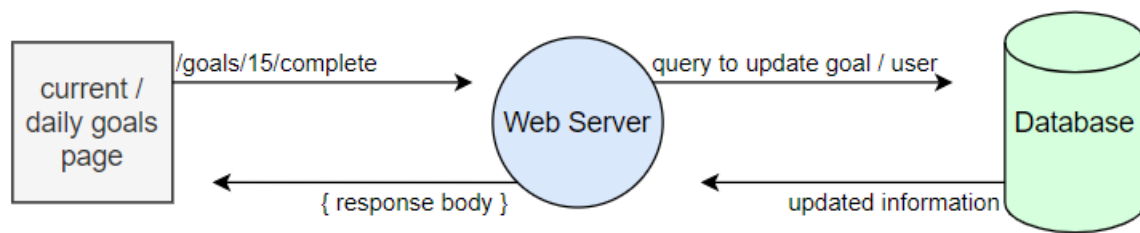


**POST goals/{goalId}/complete**

Updates the goal with the specified id to represent that it is complete. This endpoint should also calculate the points earned from completing the goal and update the user database entry accordingly.

Parameters	Description
goalId	ID of goal to update

Request	Response
To: https://goal-e/goals/15/complete	200  Body: <pre>{   "pointsAdded": 15000,   "newPointsTotal": 30000 }</pre>
To: https://goal-e/goals/15/complete If invalid goal id	400  Body: <pre>{   "error": "Goal with id: 15 not found" }</pre>
To: https://goal-e/goals/15/complete If NOT logged-in	401  Body: <pre>{   "error": "Not logged in" }</pre>



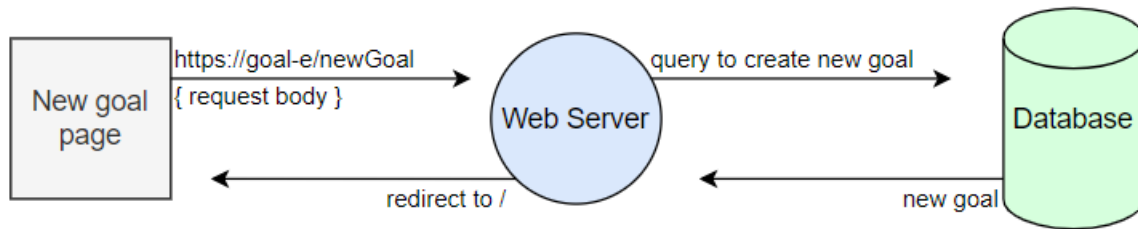
**POST /newGoal**

Creates a new goal for the logged-in user and, if successful, redirects to /.

Parameters	Description
title	Goal title
description	Goal description
deadline	Date the goal should be completed by
progress	Number between 0 and 100 indicating the progress towards goal completion
priority	Number between 1 and 3 representing the goal's priority

Request	Response
To: https://goal-e/newGoal Body: <pre>{   "title": "Walk 1000 steps",   "description": "description...",   "deadline": "11/05/2023",   "progress": 25,   "priority": 2 }</pre>	302  Redirect to /  Display success message on home page
To: https://goal-e/newGoal Body: <pre>{   "title": "invalid title",   "description": "invalid description",   "deadline": "invalid deadline",   "progress": "invalid progress",   "priority": "invalid priority" }</pre>	400  Body: <pre>{   "title": ["issue 1"],   "description": ["issue 1"],   "deadline": ["issue 1"],   "progress": ["issue 1"],   "priority": ["issue 1"] }</pre>
To: https://goal-e/newGoal  If NOT logged-in	401  Body: <pre>{   "error": "Not logged in" }</pre>

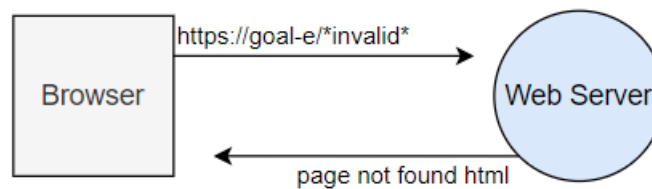
\*JSON server response can be used by user interface to display error messages without reloading



### GET /\*invalid\*

Returns the page not found html when a request is made to an invalid URL path

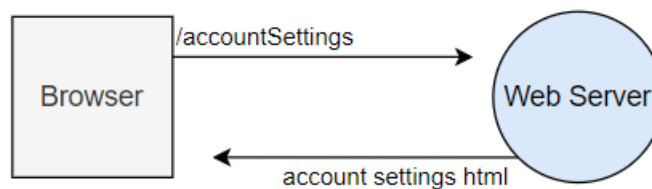
Request	Response
Browser: https://goal-e/*invalid*	404
If logged-in	Page not found html
Browser: https://goal-e/*invalid*	302
If NOT logged-in	Redirect to /login



**GET /accountSettings (stretch goal)**

Returns the html for the account settings page rendered with the logged-in user's information.

Request	Response
Browser: https://goal-e/accountSettings If logged-in	200 Account settings html
Browser: https://goal-e/accountSettings If NOT logged-in	302 Redirect to /login

**PUT /accountSettings (stretch goal)**

Updates the logged-in user's information.

Parameters	Description
username	Username string
password	Password string
confirmPassword	Password string
theme	Theme number (1 for light, 2 for dark)

Request	Response
To: https://goal-e/accountSettings Body: <pre>{   "username": "valid_username",   "password": "valid_password",   "confirmPassword": "valid_password",   "theme": 2 }</pre>	200  Body: <pre>{   "message": "successfully updated user!" }</pre>

To: https://goal-e/accountSettings Body: <pre>{   "username": "invalid_username",   "password": "invalid_password",   "confirmPassword": "invalid_password",   "theme": "invalid_theme" }</pre>	400 Body: <pre>{   "issues": {     "username": ["issue 1", "issue 2"],     "password": ["issue 1", "issue 2"],     "theme": "invalid theme"   } }</pre>
Browser: https://goal-e/accountSettings If NOT logged-in	401 Body: <pre>{   "error": "Not logged in" }</pre>

\*JSON server response can be used by user interface to update information without reloading

