Jordan Luster 7/23/25

Fitness Warriors: Service Layers

On the backend of my web application, I will be using Express with Node.js. The application will include user authentication, workout logging, adjustable stats, a progress dashboard, and scheduling features. Each of these will be represented as endpoints and follow RESTful API conventions.

1. Create new user:

Method: POST

URL: https://fitness-warrior.herokuapp.com/users

Purpose: Handles user registration. A new account is created along with initial stat entries. After submitting valid information, the user is redirected to the results page.

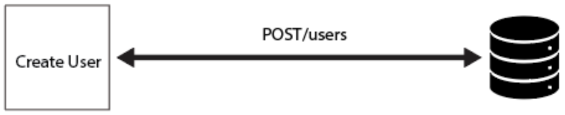
Example Request:

```
POST /users HTTP/1.1

Host: https://fitness-warrior.herokuapp.com/users

{
        "name": "Jason Willis",
        "email": "jasonwillis@yikes.com",
        "password": "rockstar27",
        "goals": "lose weight",
        "schedule": "early morning",
        "activity level": "not active",
        "height": "5' 7\"",
        "current weight": "250.5",
        "goal weight": "200.0",
        "gender": "male",
        "DOB": "10-20-1995"
}
```

Success response:

```
{
        "message1":  "Stats added successfully!. Account created successfully"
        "recommended workout plan": "WorkoutPlan"
        }
        Error Response:
        {
        "error":  "Password needs eight characters",
        "code”: 400,
}
```

Diagram:



2. Check Stats:

Method: GET

URL: https://fitness-warrior.herokuapp.com /users/:id/stats

Purpose: Retrieves the user's current stats. This data appears when the user selects the stats option on the hub page and can be edited directly on that page.

Example Request:

GET /users HTTP/1.1

Host: http s://fitness-warrior.herokuapp.com /users/:id/stats

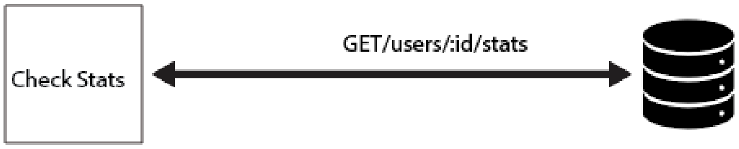
Success Response:

```
{
        "name": "Jason Willis",
        "goals": "lose weight",
        "schedule": "early morning",
        "activity level": "not active",
        "height": "5' 7\"",
        "current weight": "250.5",
        "goal weight": "200.0",
        "gender": "male",
        "DOB": "10-20-1995"
}
```

Error Response:

```
{
        “Code”: 400,
        “details”: “The requested information cannot be found”
}
```

Diagram:



3. Log Workout:

Method: POST

URL: https://fitness-warrior.herokuapp.com/logs

Purpose: Records a completed workout. This is triggered when the user logs a workout session, and the system confirms it was successfully submitted.

Example Request:

```
POST /users HTTP/1.1
Host: https://fitness-warrior.herokuapp.com/logs
{
        "workout name": "bench press",
        "sets": "3"
        "reps": "20",
        "weight": "135.5 lbs"
}
```

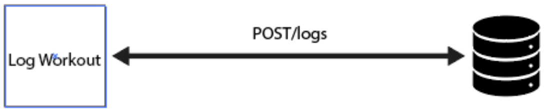
Success Response:

```
{

        “message”: “Workout logged successfully”,

}
```

Error Response:

```
{

        “message”: “number of reps required”,

}
```

Diagram:



4. Create a Workout

Method: POST

URL: https://fitness-warrior.herokuapp.com/workouts

Purpose: - Allows the user to manually create and store a new workout plan. This is useful for customizing workouts on the workout log page.

Example Request:

```
POST /users HTTP/1.1

Host: example.com

{

        "workout type": "weight lifting",

        "workout name": "bench press",

        "intensity": "vigorous effort",

        "time duration": "60.0 min"

}
```

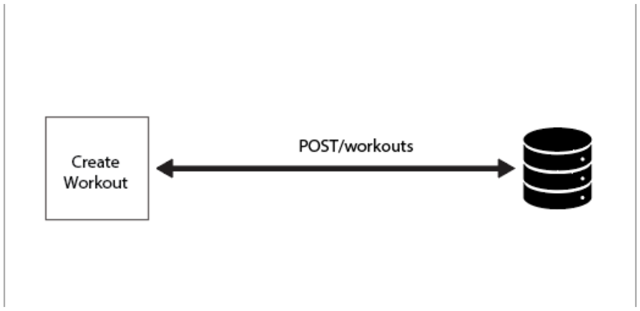
Success Response:

```
{

        “message”: “Workout created successfully”

}
```

Error Response:

```
{
        “error”: “cannot be blank”,
}
```

Diagram:



5. Check Progress:

Method: GET

URL: https://fitness-warrior.herokuapp.com/users/:id/progress

Purpose: - Displays a summary of progress data such as weight changes, completed workouts, and milestones. This information is used to generate visual progress insights on the hub page.

Example Request:

GET /users HTTP/1.1

Host: https://fitness-warrior.herokuapp.com /users/:id/progress

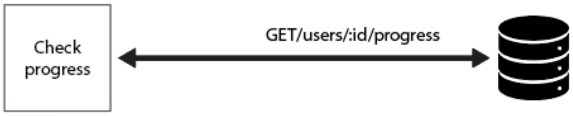
Success Response:

```
{
        "user_id": 1091,
        "progress": {
         "start_weight": 250.5,
        "goal_weight": 200.0,
        "current_weight": 235.2,
         "workouts_completed": 42,
        "milestones": ["Lost 10 lbs", "Hit 30 workout streak"],
        "updated_at": "2025-07-15T02:30:00Z"
},
status": "success"
}
```

Error Response:

```
{
        "error": "Progress data not found",
        "code": 404,
        "details": "No progress records available for the given user ID",
}
```

Diagram:



6. Delete Workout:

    Method: DELETE

    URL: https://fitness-warrior.herokuapp.com/workouts/:id

Purpose: - Removes a previously logged workout entry. This option is available when users want to delete incorrect or outdated workout records.

Example Request:

DELETE /users HTTP/1.1

Host: https://fitness-warrior.herokuapp.com/workouts/:id

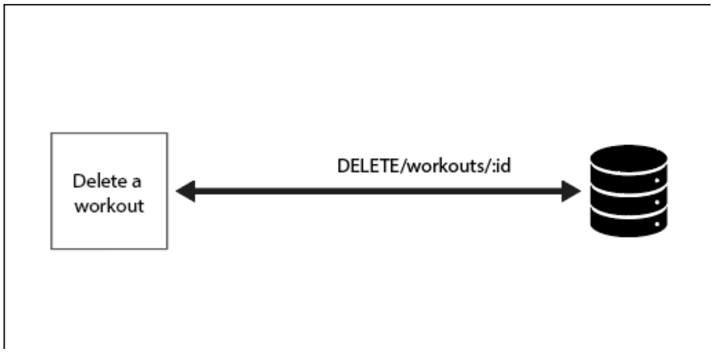
Success Response:

```
{

        "message": "Workout deleted successfully",

         "workout_id": 872,

        "status": "success",

}
```

Error Response:

```
    {

      "error": "Workout not found",

      "code": 404,

      "details": "No workout entry exists with the specified ID",

    }
```

Diagram:



7. Change Stats:

Method: PATCH

URL: https://fitness-warrior.herokuapp.com /users/:id /stats

Purpose: Allows a user to update specific profile statistics such as weight, goals, or activity level. This endpoint is triggered when the user selects the "Quick Stats" option on the dashboard and submits updated information. The request modifies only the changed fields in the user's stored profile.

Example Request:

```
PATCH /users HTTP/1.1

Host: https://fitness-warrior.herokuapp.com /users/:id/stats

{

        "name": "Jason Willis",

        "goals": "lose weight",

        "schedule": "early morning",

        "activity level": "not active",

        "height": "5' 7\"",

        "current weight": "250.5",

        "goal weight": "200.0",

        "gender": "male",

        "DOB": "10-20-1995"
}
```

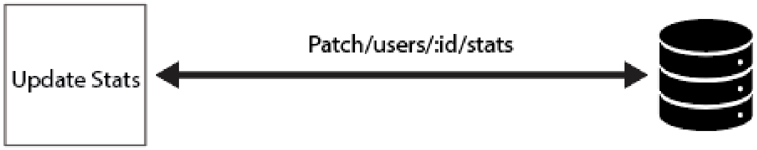
Success Response:

```
{
        “message”: “Stats updated successfully”,
        “updated_fields”: [“current_weight”, “activity_level”, “goal_weight”],
        “status”: “success”
}
```

Error Response:

```
{
        “Code”: 400,
        “details”: “Missing required field: ‘goal_weight’”
}
```

Diagram:



8. Reschedule workout:

Method: PATCH

URL: https://fitness-warrior.herokuapp.com /users/:id /reschedule

Jordan Luster 7/23/25

Purpose: User misses a day of their recommended workout. This means they simply did not log the workout. The system gives them a chance to log that workout or push the schedule back. It will show up as a pop up on the dashboard.

Example Request:

```
PATCH /users HTTP/1.1

Host: https://fitness-warrior.herokuapp.com /users/:id/workouts/reschedule

{

        “Message”: “You did not log your workout today. Would you like to log your workouts
        now?"

}
```

Success Response:

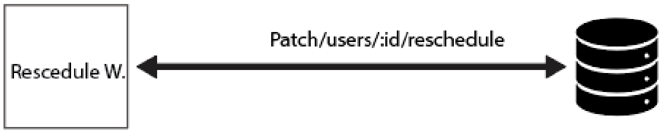
```
{

        "message": "Workout rescheduled to next available slot",

         "status": "success",

          "workout_id": 872}
```

Error Response:

```
{

  "error": "You did not log anything?",

  "code": 409

}
```

Jordan Luster 7/23/25

Diagram:


Rescedule W.
Patch/users/:id/reschedule