End-to-End Scenario: Scheduling a Recreational League with Constraints

Description of Setup

Imagine a city Parks & Recreation department needs to schedule a small recreational soccer league composed of 8 teams. They have one venue with 4 fields available uniformly every day from 9:00 AM to 5:00 PM. The league's requirements are:

1. Each team must face every other team exactly once (so each team plays 7 games, since there are 8 teams).
2. No team should play more than one game per day.
3. Games last exactly 2 hours.
4. The schedule should have no overlapping games on the same field and must respect the venue's availability.

The input data (team.csv, league.csv, venue.csv) for this scenario is placed in ./data/case1/. The code has been modified to enforce the once-per-day rule for all teams and to schedule as many games as needed without conflicts.

Step-by-Step Execution Procedure

1. **Prepare the Environment**:

   Ensure you have run the provided scripts:

   bash

   Copy code

   ```
   ./bin/build-env
   source ./bin/run-env
   ```

   ○

2. **These scripts set up the Python environment and dependencies needed to run the scheduler and the web server.**
3. **Launch the Web Server**:

   Start the FastAPI server by running:

   bash

   Copy code

   ```
   ./bin/launch
   ```

4. **This command will start the local server on https://localhost:8000/.**

   Login:

   email: user@example.com

   password: password

   - This will log you into the SPORTS scheduling interface.
5. **Navigate to the Scheduler Page**:
   - After logging in, select the "Schedule" option from the menu.
   - You will see buttons for "Case 1" through "Case 8" and "Generated".
6. **Select Case 1**:
   - Click on the "Case 1" button to run the scheduler for our 8-team recreational league scenario.
   - The system will run the scheduling logic based on ./data/case1/ input files.
7. **Observe the Results**:
   - If the scheduling is successful, you will see a message like "Test case1 passed!"
   - The resulting schedule.csv and schedule.json files will be generated in ./data/case1/.
8. **Examine the Output Files**:
   - Locate data/case1/schedule.csv.
   - Open it to confirm the presence of headers: team1Name, team2Name, week, day, start, end, season, league, location.
9. **Verify the Constraints**:

   - **No Overlaps on Same Field**: Check that no two games share the same field and time slot. For example, if one game is on Field #1 from 9:00–11:00 AM, there should be no other game on Field #1 during that interval.
   - **Each Team Once Per Day**: Pick any team (e.g., "Team 1") and verify that across all rows in a single day, Team 1 appears in only one match.
   - **All Matchups Occur**: Ensure each team faces the 7 other teams exactly once. Since we have 8 teams, you should find 7 games for each team spread across multiple weeks/days.
   - **2-Hour Slots**: Confirm start/end times always differ by 2 hours (e.g., 9:00 to 11:00, 11:00 to 13:00 (1:00 PM), etc.).
   - **No More Than One Game Per Team Per Day**: For each day and for each team, ensure the team_daily_count logic was enforced by checking that no team name appears more than once on the same day.

10. **Traceability to Requirements**:
    - **Requirement: Each team plays every other team once**: By counting the occurrences of each team, you will see each has faced all others in the schedule.
    - **Requirement: No team plays more than once a day**: A quick inspection of the schedule.csv will show that the daily limit was respected.
    - **Requirement: No overlaps and 2-hour increments**: The interval trees and time slot checks ensure no overlapping fields and each slot lasts exactly 2 hours.
    - **Requirement: Fields and times fit within the venue's 9:00–17:00 availability**: All scheduled games should fall within these hours, as per the input data in venue.csv.

Verification Checks

- After inspecting schedule.csv, you confirm:
  1. **Headers present**: team1Name, team2Name, week, day, start, end, season, league, location are correct.
  2. **Correct number of games (28)**: With 8 teams, each playing 7 games, you have 28 entries total.

3. **Once-per-day per team**: No team appears twice on the same season, week, day.
4. **No overlapping fields**: No two games share the same field and time interval.
5. **Time Slots are 2 hours**: Differences between end and start are always 2 hours.

If all these checks pass, the scenario is validated end-to-end.

Traceability to Requirements

- **R1: Every team faces all others once**: Verified by checking matchups in schedule.csv.
- **R2: No team more than once per day**: Confirmed by examining the daily distribution of games per team.
- **R3: No overlapping fields and 2-hour increments**: Confirmed by time and field intervals.
- **R4: Fits venue availability**: All scheduled start/end times fall within provided daily availability windows.

All these requirements map to the constraints mentioned in the scenario, ensuring the end-to-end process from input preparation to output verification aligns with the league’s stated needs.