langchain-ai / ff-take-bot

Code Issues Pull requests Actions Projects Security Insights

9 stars 3 forks 3 watching Branches Activity Custom properties Tags

Public repository

1 Branch 0 Tags Go to file t Go to file + Add file Code

rlancemartin Update README.md 726380d · 7 months ago

ntbk	Update README	7 months ago
.env.example	Add take-bot	9 months ago
README.md	Update README.md	7 months ago
langgraph.json	Add take-bot	9 months ago
populate_env.sh	Update	7 months ago
requirements.txt	Add take-bot	9 months ago
take_bot.py	Add cron	7 months ago

README

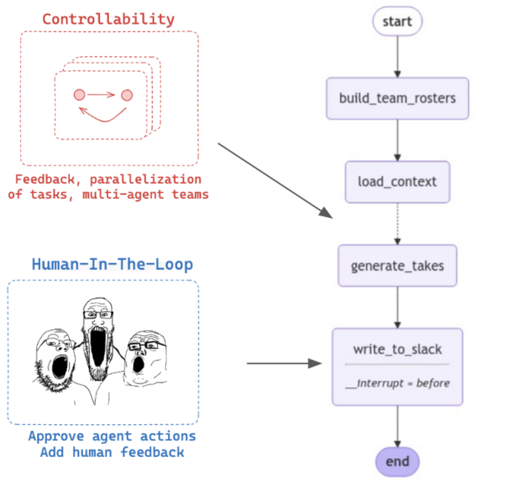
Fantasy Football Take Bot

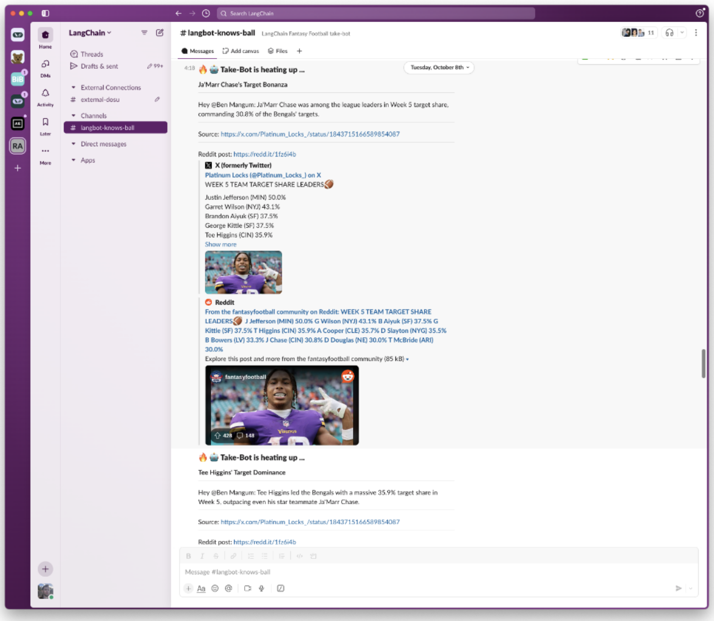
Ambient agents represent a shift from chat-based AI interactions to AI that works quietly in the background, monitoring and analyzing information without requiring constant user attention. While chat interfaces limit us to one task at a time, ambient agents can handle multiple tasks simultaneously, working diligently without the pressure of real-time response expectations.

This repository demonstrates a fun internal example we use at LangChain: a Fantasy Football "Take Bot" that runs in the background to keep our league engaged. This agent:

1. Scrapes recent posts from the Reddit `fantasyfootball` sub
2. Gets our Fantasy Football league data from the ESPN API
3. Generates short-summaries (or "takes") for each Fantasy team manager based on their roster
4. Publishes the takes to a Slack channel
5. The app is deployed to LangGraph Platform and scheduled to run once a day

Instead of managers having to actively seek out fantasy football insights, this ambient agent does the work for them, delivering daily insights directly to Slack. It's a practical example of moving from "human-in-the-loop" to "human-on-the-loop" - the agent works autonomously, but all its steps are observable and the output can be reviewed and adjusted as needed.





langbot-knows-ball LangChain Fantasy Football take-bot

4:18 🔥 🤖 Take-Bot is heating up ...

Ja'Marr Chase's Target Bonanza

Hey @Ben Mangum: Ja'Marr Chase was among the league leaders in Week 5 target share, commanding 30.8% of the Bengals' targets.

Source: https://x.com/Platinum_Locks_/status/1843715166589854087

Reddit post: https://redd.it/1fz6i4b

X (formerly Twitter)
Platinum Locks (@Platinum_Locks_) on X
WEEK 5 TEAM TARGET SHARE LEADERS🏈

Justin Jefferson (MIN) 50.0%
Garret Wilson (NYJ) 43.1%
Brandon Aiyuk (SF) 37.5%
George Kittle (SF) 37.5%
Tee Higgins (CIN) 35.9%
Show more

Reddit
From the fantasyfootball community on Reddit: WEEK 5 TEAM TARGET SHARE LEADERS🏈 J Jefferson (MIN) 50.0% G Wilson (NYJ) 43.1% B Aiyuk (SF) 37.5% G Kittle (SF) 37.5% T Higgins (CIN) 35.9% A Cooper (CLE) 35.7% D Slayton (NYG) 35.5% B Bowers (LV) 33.3% J Chase (CIN) 30.8% D Douglas (NE) 30.0% T McBride (ARI) 30.0%
Explore this post and more from the fantasyfootball community (85 kB)

fantasyfootball
428 148

🔥 🤖 Take-Bot is heating up ...

Tee Higgins' Target Dominance

Hey @Ben Mangum: Tee Higgins led the Bengals with a massive 35.9% target share in Week 5, outpacing even his star teammate Ja'Marr Chase.

Source: https://x.com/Platinum_Locks_/status/1843715166589854087

Reddit post: https://redd.it/1fz6i4b

Data Sources

Reddit

The Take Bot scrapes recent posts from the [Reddit `fantasyfootball` subreddit]() to gather current fantasy football discussions and trends. To set up Reddit API access:

1. Go to [Reddit's App Preferences]()
2. Click "Create Application" or "Create Another Application"
3. Fill in the following:
   - Name: `fantasy-football-take-bot` (or any name you prefer)
   - Select "script" as the application type
   - Description: Optional, but helpful for remembering the app's purpose
   - About URL: Can be left blank for personal use
   - Redirect URI: Use `http://localhost:8080`
4. Click "Create app"
5. Once created, you'll see your credentials:
   - Client ID: Found under your app name
   - Client Secret: Listed as "secret"

Add the following credentials to your environment:

- `REDDIT_CLIENT_ID`
- `REDDIT_CLIENT_SECRET`

ESPN API

Our Fantasy Football league using the ESPN app. We used the espn-api package to access the league data.

Add add the following credentials to your environment:

- `ESPN_LEAGUE_ID` : get this from your ESPN url, `leagueId` : `https://fantasy.espn.com/football/team?leagueId=xxx&teamId=y&seasonId=2024`
- `ESPN_S2` : get this as shown here
- `ESPN_SWID` : get this as shown here

Publishing to Slack

Since ambient agents work best when integrated into existing workflows, we publish the agent's insights directly to Slack where our team already communicates. This allows managers to passively monitor the agent's analysis while going about their day. To set this up:

1. Go to https://api.slack.com/apps
2. Click "Create New App"
3. Choose "From scratch"
4. Name your app (e.g., "Take Bot") and select your workspace
5. Once created, go to "Incoming Webhooks" in the left sidebar
6. Toggle "Activate Incoming Webhooks" to On
7. Click "Add New Webhook to Workspace"
8. Choose the channel where you want the messages to appear
9. Copy the "Webhook URL" that's generated

Add add webhook URL credentials to your environment:

- `TAKE_BOT_SLACK_URL`

Testing with the notebook

Create your environment and run the notebook to test the graph and your Slack connection.

```
$ python3 -m venv take-bot-env
$ source take-bot-env/bin/activate
$ pip install -r requirements.txt
$ jupyter notebook
```

Running Studio

You can use the LangGraph Studio desktop app to run the agent locally (on your own machine).

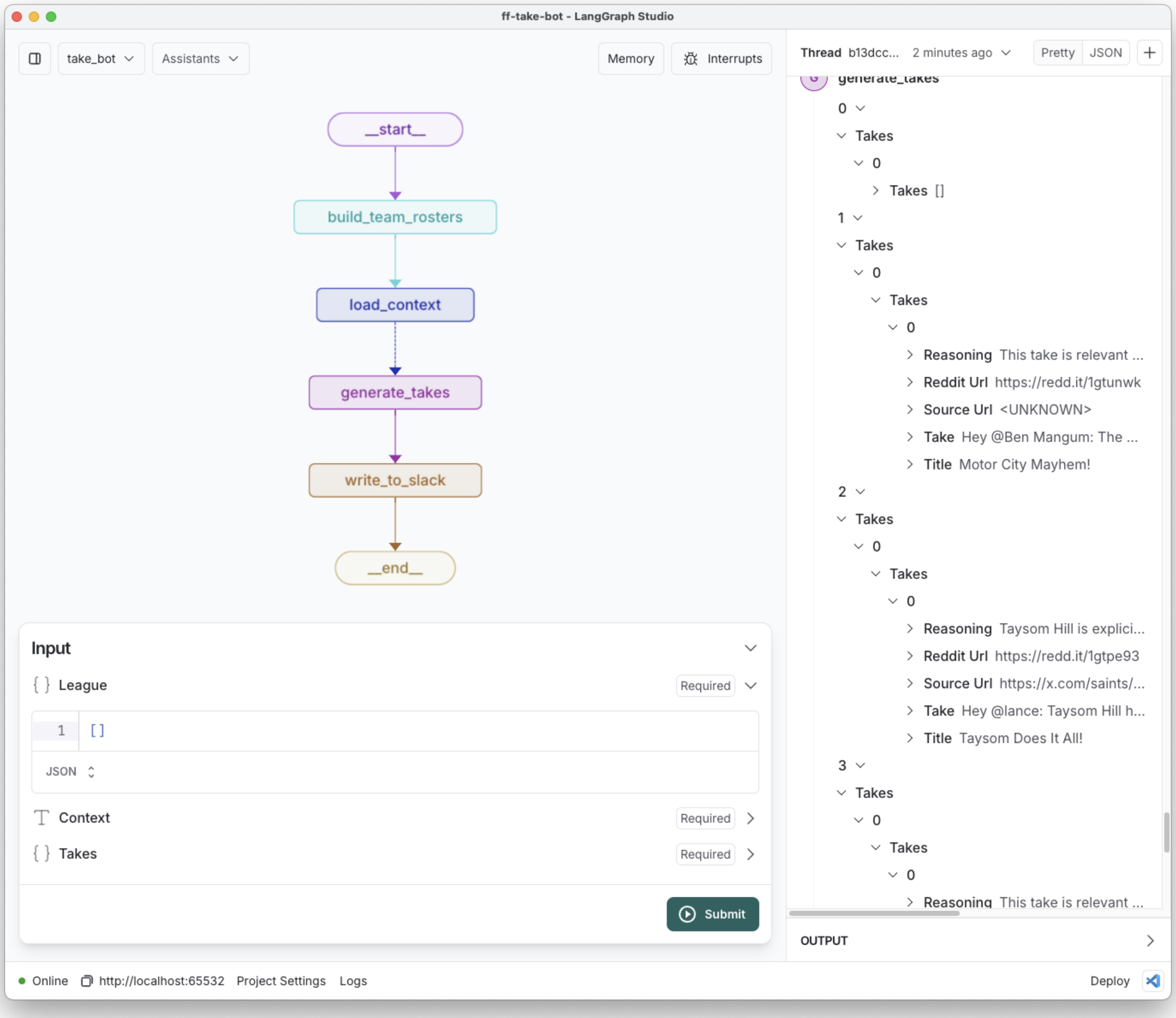
To do this, first download the desktop app and have Docker Desktop running.

Generate your `.env` file with the necessary credentials:

```
$ ./populate_env.sh
```

Load this folder in the Studio app to launch it.

Simply run `submit` with default values (`[]`) passed to the input state; the app will populate these as it runs.



```
generate_takes
  0
  Takes
    0
      Takes []
  1
  Takes
    0
      Takes
        0
          Reasoning This take is relevant ...
          Reddit Url https://redd.it/1gtunwk
          Source Url <UNKNOWN>
          Take Hey @Ben Mangum: The ...
          Title Motor City Mayhem!
  2
  Takes
    0
      Takes
        0
          Reasoning Taysom Hill is explici...
          Reddit Url https://redd.it/1gtpe93
          Source Url https://x.com/saints/...
          Take Hey @lance: Taysom Hill h...
          Title Taysom Does It All!
  3
  Takes
    0
      Takes
        0
          Reasoning This take is relevant ...
```

```
1 []
```

Hosted Deployment

We deployed our app to LangGraph Cloud, a managed service for running LangGraph graphs. This makes it easy to set up a scheduled job to run the graph on a regular basis (e.g., daily) and publish the results to Slack. As shown in the notebook, you can use the LangGraph Python SDK to create a scheduled job:

```
from langgraph_sdk import get_client

# URL from our LangGraph Cloud deployment
deployed_url = "https://ff-take-bot-deployment-
f4901b2dbda85d9787dac18e2a977956.default.us.langgraph.app"
client = get_client(url=deployed_url)

# An assistant ID is automatically created for each deployment
await client.assistants.search(metadata={"created_by": "system"})

# Set the assistant ID you want to create a cron job for
```

```
assistant_id = 'xxx'

# Use she SDK to schedule a cron job e.g., to run at 1:00 PM PST (21:00 UTC) every day
cron_job_stateless = await client.crons.create(
    assistant_id,
    schedule="0 21 * * *",
    input={"league": []}
)
```

Releases

No releases published

Packages

No packages published

Languages

● **Jupyter Notebook** 94.7% ● **Python** 5.2% ● **Shell** 0.1%