



D&D Beyond: Random NPC Generator

Milestone 2

Brion Gahl
Prem Rana
Caleb Rapsilber



PROJECT SUMMARY

❑ Key Figures

- ❑ Patrick Backmann (Project Manager)
- ❑ Jason Bush (Software Engineer)
- ❑ Ben Dixon (Backend Software Engineer)

❑ NPC generator

- ❑ Importing of characters to NPCs or Randomly generating a NPC
- ❑ NPC collections
- ❑ All D&D source data as well as character / NPC data

❑ Workload Distribution

- ❑ Prem: Backend / Database / Flask
- ❑ Brion: NPC Builders & Sheets / React Implementation / Backend
- ❑ Caleb: Design / Styling / NPC Builders & Sheets

PROGRESS SINCE MILESTONE ONE

Frontend

- ❑ Allowed React components to interface with backend
- ❑ Created NPC sheet page to display character information
- ❑ Created builders to allow creation of NPCs

Backend

- ❑ Connected to Simple Character Data Service to extract character info
- ❑ Stored User and Character data in preliminary database
- ❑ Integrated with Front-End to populate NPC information

DESIGN PROCESS

❏ End User

❏ Familiarity

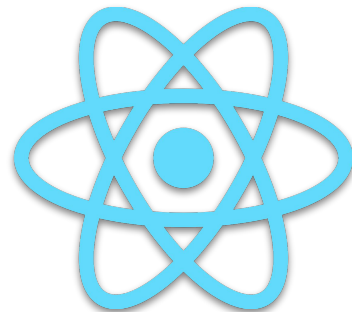
- ❏ Layout, Interactions, Styling

❏ Functionality

- ❏ Focus on GMs
- ❏ Use cases: Importing, Random, Scalability, Accessibility

❏ Guidance from Pat

- ❏ Most impactful problem to user, not solution
- ❏ Multiple use cases
- ❏ Expansion (like a lego kit)



FINAL SPECS



Features:

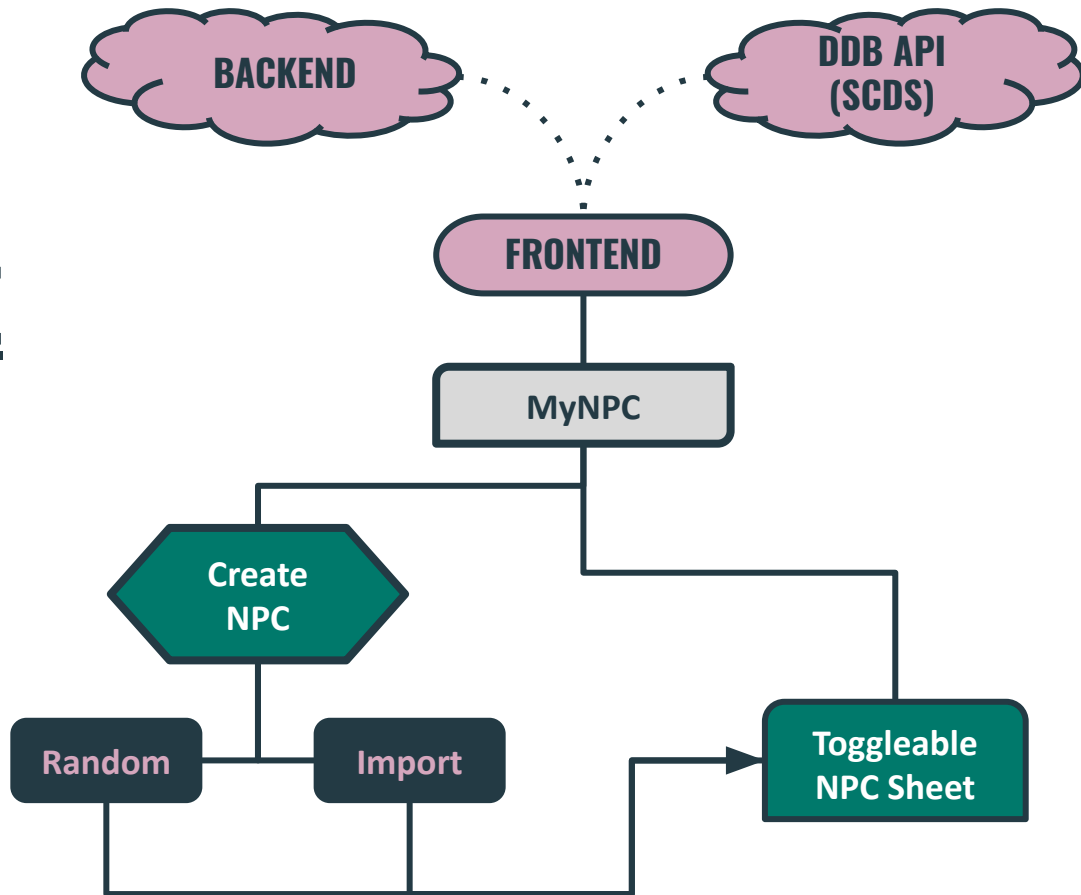
- ☐ Interface with DDB APIs
- ☐ Importing Characters
- ☐ Random Generation of NPC
- ☐ Displaying data in a traditional NPC Sheet
- ☐ Displaying data in a DDB style character sheet
- ☐ Store imported characters, attached to users



Built on:

- ☐ React Stack
- ☐ Python Web-Server
- ☐ SQLite Database

PROJECT ARCHITECTURE



WORK IN PROGRESS / FUTURE

- ❑ Further backend support
 - ❑ Creation of Flask server to be used for React-Router use
- ❑ Touch up on features / interactions
- ❑ Further/finish design and styling
- ❑ Integration, testing, and potential staging through D&D Beyond
- ❑ STRETCH GOAL: Group page to create a group of NPCs



DEMO SHOWCASE