

Guidance for Students: D&D Character API

Introduction

Welcome to the `virtual-robot-maze` project, adapted to create and interact with a Dungeons & Dragons (D&D) character! In this project, you will use Rust to interact with a black box API that simulates a D&D character sheet. Your goal is to call functions from the API to retrieve character attributes like name, class, ancestry, and stats like strength and intelligence.

Getting Started with GitHub and Codespaces

1. Fork the Repository

- Go to the main repository on GitHub.
- Click the **Fork** button at the top right to create your own copy of the project.
- You can find the forked repository under your GitHub profile.

2. Open GitHub Codespaces

- In your forked repository, click the **Code** button and select **Codespaces**.
- If you don't have a codespace yet, create one by clicking **New Codespace**.
- Once it loads, you'll have a development environment with Rust pre-installed, ready for coding.

Understanding the Project Structure

The `virtual-robot-maze` repository contains several Rust projects and tasks:

- **Basic Programming Tasks:** These tasks, found in the `basic-programming` directory, introduce Rust programming concepts.
- **D&D Character Tasks:** These tasks, located in the `src` directory, allow you to interact with the black box API and build a D&D character.

Switching Between Projects

You'll need to switch between different Rust projects for each task. Here's how:

1. To work on basic programming tasks:

- Navigate to the `basic-programming` directory:

```
1 cd ~/workspaces/virtual-robot-maze/basic-programming/basic_syntax
```

- Use `cargo run` to run the code for basic programming tasks.

2. To work on D&D character tasks:

- Navigate to the `src` directory where the D&D character code resides:

```
1 cd ~/workspaces/virtual-robot-maze/src
```

- Use `cargo run` to execute your code and interact with the D&D character API.

Interacting with the D&D Character API

The API allows you to query various attributes of your character. Here are some key functions:

- `get_character_name()`: Returns the character's name.
- `get_character_class()`: Returns the character's class, like "Wizard" or "Fighter".
- `get_character_ancestry()`: Returns the character's ancestry, like "Human" or "Elf".
- `get_strength()`, `get_dexterity()`, etc.: Returns attribute scores.

General Workflow Tips

- Always use the terminal to navigate between directories before running commands.
- Check your directory with `pwd` before running commands to ensure you're in the correct location.
- Save your changes frequently and use `git commit` and `git push` to keep track of your work.

Next Steps

Once you understand the structure and how to switch between projects, you can start on the tasks. Refer to specific task documents for instructions on querying character details and calculating modifiers.