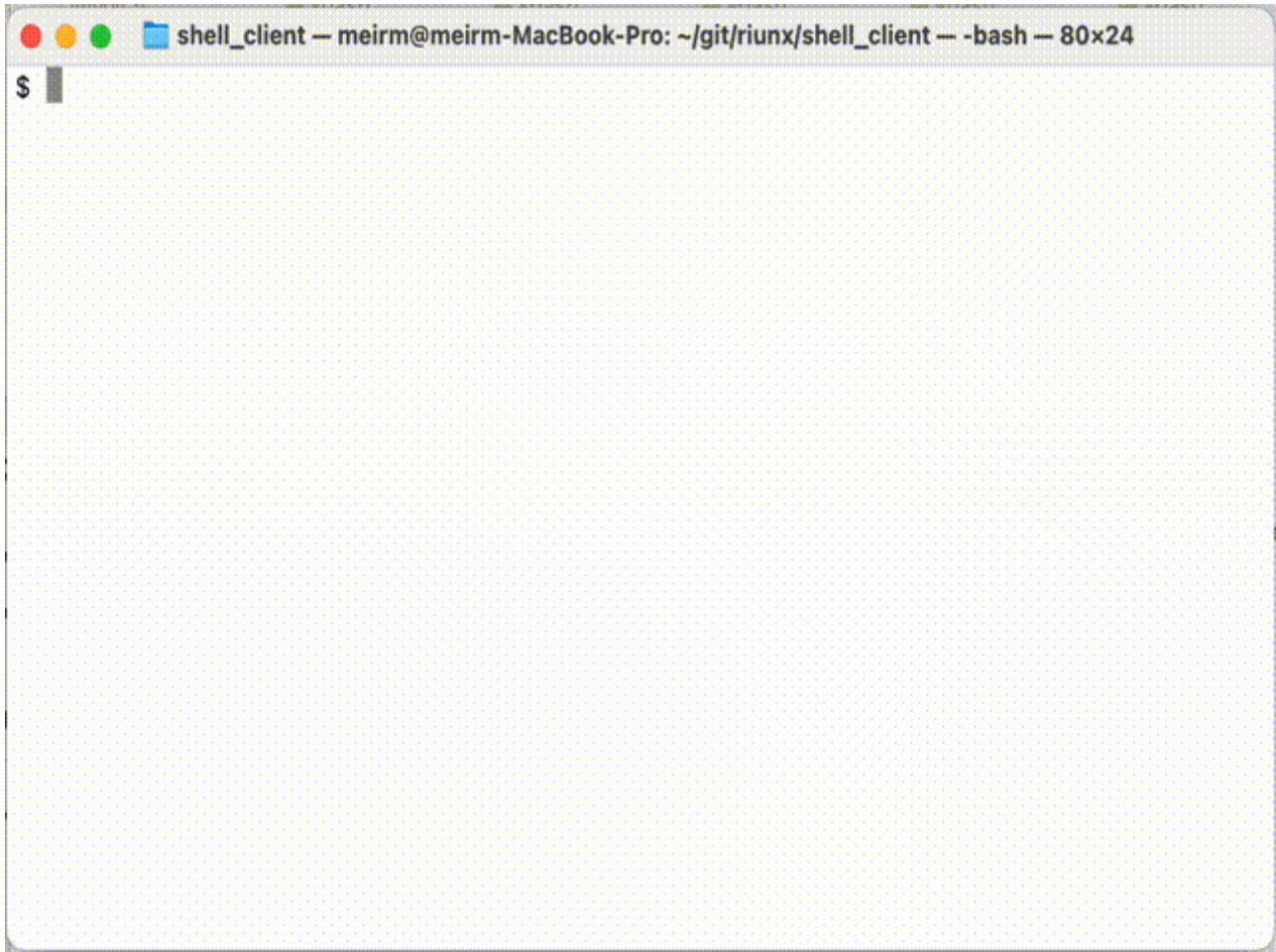


shell_client - Vanilla Shell Client Framework

The Vanilla Shell Client Framework provides a foundational command-line interface for building custom shell applications. This framework is designed to be lightweight and extendable, offering basic commands and a structure for developers to add specific functionalities tailored to their needs.



Features

- **Basic Commands:** Includes essential commands such as `exec`, `help`, and `exit` to perform operations within the shell environment without leaving the session.
- **Extendable:** Designed for developers to build upon, supporting easy integration of additional commands and functionalities.

Getting Started

To get started with the Vanilla Shell Client Framework, clone the repository and switch to the main branch.

```
git clone https://github.com:meirm/shell_client.git
cd shell_client
git checkout main
```

Prerequisites

- Python 3.x
- Additional dependencies (if any) listed in `requirements.txt`. Install them using:

```
pip install -r requirements.txt
```

Running the Shell

To start the shell session, run:

```
python client/main.py
```

Using the DRF Branch

For developers working with Django Rest Framework (DRF), the `drf` branch of this repository includes implementations of HTTP request methods (`GET`, `POST`, `PATCH`, `DELETE`, `HEAD`, `OPTIONS`, `PUT`) to interact with DRF-based APIs.

Switching to the DRF Branch

To use the DRF-specific features, switch to the `drf` branch after cloning the repository:

```
git checkout drf
```

DRF Branch Features

- **HTTP Request Commands:** Introduces commands corresponding to HTTP methods for API interaction.
- **Authentication Support:** Implements mechanisms for handling authentication with APIs, including token handling.

Extending the Framework

Developers are encouraged to extend the framework by adding new commands or modifying existing ones to fit their requirements. To add a new command:

1. Define a new method in your shell class following the naming convention `do_<command>`.
2. Implement your command logic within this method.

Refer to the [cmd module documentation](#) for detailed guidance on extending and customizing the command interface.

Contributing

Contributions to both the main and DRF branches are welcome. Please follow the standard Git workflow for contributions:

1. Fork the repository.
2. Create a new feature branch from `main` or `drf` depending on the contribution context.
3. Commit your changes.
4. Push your branch and create a pull request against the relevant branch.

License

MIT License (c) 2024 Meir M. All rights reserved.