Source Code

Warring
The content of this page is still in progress and needs some tweaking.

Introduction

Welcome to the Unicis Source Code Handbook. This guide provides an overview of how we manage our source code across different repositories, both private and public. It covers the structure, guidelines, and best practices for contributing to the Unicis Platform.

Repository Overview

Unicis manages two separate sets of repositories:

- **Private Repositories**: Hosted on GitLab, containing proprietary source code for Unicis Platform Business Edition, Unicis Apps, website, and this handbook.
- **Public Repository**: Hosted on GitHub, containing the open-source code for the Unicis Platform Community Edition.

Private Repositories on GitLab

Our private repositories are used for internal development and contain sensitive, proprietary code. These repositories are not open to the public.

Unicis Platform Business Edition

The Business Edition of the Unicis Platform includes enterprise features and custom integrations.

Repository Structure (Next.js Tech Stack)

```
unicis-platform
├── components
├── data
├── hooks
├── lib
└── locales/en
└── models
└── pages
└── prisma
└── public
└── style
└── test/e2e/auth
└── types
```

Access

Access to this repository is restricted to authorized personnel. Ensure you have the necessary permissions to clone and contribute to this repository.

Development Guidelines

- **Branching Strategy**: Use a Git branching strategy such as GitFlow.
- **Commit Messages**: Follow the commit message guidelines to ensure clarity and consistency.
- **Code Reviews**: All changes must go through a code review process before being merged.

Unicis Apps (Atlassian Forge Framework)

This repository contains the source code for various applications developed using the Atlassian Forge framework.

Repository Structure

```
unicis-apps
├── app1
├── app2
├── common
└── README.md
```

Development Guidelines

- **Modular Structure**: Each application should be developed as a separate module within the repository.
- **Dependency Management**: Ensure all dependencies are properly documented and managed.

Website (Docusaurus)

This repository hosts the source code for the Unicis website.

Repository Structure

```
website
├── blog
├── docs
├── resources
├── src
├── static
├── versioned_docs
├── versioned_sidebars
```

Deployment

- **Website**: Deployed using Docusaurus framework.

Handbook (Hugo CMS)

This repository hosts the source code for the Unicis handbook.

Repository Structure

```
unicis-handbook
├── archetypes
├── assets
├── content
├── layouts/partials/docs
├── layouts/_gen/assets
├── static
├── themes
```

Deployment

- **Handbook**: Managed and deployed using DokuWiki.

Public Repository on GitHub

Our public repository on GitHub is open for contributions from the community. This repository contains the Community Edition of the Unicis Platform.

Community Edition

The Community Edition is a feature-limited version of the Unicis Platform, designed to be open and extensible.

Repository Structure

```
unicis-platform-ce
├── components
├── data
├── hooks
├── lib
└── locales/en
└── models
└── pages
└── prisma
└── public
└── style
```

```
└── test/e2e/auth
└── types
```

Access

The repository is public and can be accessed by anyone at: Unicis Platform Community Edition.

Contribution Guide

We welcome contributions from the community. Please follow the guidelines outlined in the Contribution Guide.

Key Points

- **Fork the Repository**: Create a fork of the repository to make your changes.
- **Create a Branch**: Use a descriptive branch name for your feature or bug fix.
- **Pull Requests**: Submit a pull request with a detailed description of your changes.
- **Code of Conduct**: Adhere to the community Code of Conduct.

Best Practices

- **Consistent Coding Standards**: Follow the coding standards specified in the repository.
- **Documentation**: Keep documentation up-to-date with code changes.
- **Testing**: Write unit and integration tests for your code.
- **Security**: Regularly review and update dependencies to address security vulnerabilities.

Troubleshooting

Common Issues

- **Access Issues**: Ensure you have the correct permissions for private repositories.
- **Build Failures**: Check CI/CD pipeline logs for errors and address them accordingly.
- **Merge Conflicts**: Resolve conflicts by pulling the latest changes and merging them into your branch.

Debugging Tips

- **Logs**: Review logs for any runtime errors or warnings.
- **Documentation**: Refer to repository documentation and community forums for solutions.

Resources

- **GitLab Documentation**: GitLab Docs
- **GitHub Documentation**: GitHub Docs
- **Unicis Community**: Unicis Community

By following this handbook, you will be well-equipped to manage and contribute to the Unicis Platform effectively. For any further assistance, please reach out to the DevOps team or consult the provided resources.

From:
https://handbook.unicis.tech/ - **Unicis Handbook**

Permanent link:
https://handbook.unicis.tech/pub/development/source_code

Last update: **28.09.2024 15:59**