



**CelebManager**

**Developer Guide**



Product of CS2103JAN2018-W14-B4

Last Updated: 27 March 2018

**Table of Contents**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | Introduction | | | | | **3** |
| **2** | Setup | | | | | **4** |
|  | **2.1** | Prerequisites | | | | **4** |
|  | **2.2** | Local Project Setup | | | | **5** |
|  | **2.3** | Setup Verification | | | | **5** |
|  | **2.4** | Project Configurations | | | | **6** |
|  |  | **2.4.1** | | Coding Style Configurations | | **6** |
|  |  | **2.4.2** | | Documentation Configurations | | **6** |
|  |  | **2.4.3** | | Continuous Integration (CI) Configurations | | **7** |
| **3** | Design | | | | |  |
|  | **3.1** | | Software Architecture | | |  |
|  |  | |  | |  |  |
|  |  | |  | |  |  |
|  |  | |  | |  |  |
|  |  | |  | |  |  |
|  |  | |  | |  |  |
|  |  | |  | |  |  |
| **4** | Frequently Asked Questions | | | | |  |
| **5** | Command Summary | | | | |  |

1 Introduction

**CelebManager** is a software application that aims to allow celebrity managers to maintain schedule of celebrities under them. CelebManager is optimized for users who **prefer to work with a Command Line Interface** (CLI) while still having a Graphical User Interface (GUI) for visual feedback.

In this software, the users should be able to:

* Manage contacts
* Manage appointments
* Manage calendars
* Plan efficient routes

This developer guide aims to allow you to contribute to this project. Information on how the software is designed and implemented, and how the software should be developed and maintained is in this documentation to help you contribute to this project.

2 Setup

This section will discuss the setting up of project for development.

2.1 Prerequisites

There are two prerequisites before you can work on this software.

1. Java Development Kit (JDK)

The Java programming language is used in this project. To be able to work with this project, you will need to have JDK version 1.8.0\_60 installed.

You can download JDK [here](http://www.oracle.com/technetwork/java/javase/downloads/index.html).

|  |  |
| --- | --- |
| Lightbulb | Some components of this software will not work with earlier versions of Java 8. |

1. IntelliJ Integrated Development Environment (IDE)

This software is developed as a Gradle project, which requires you to work on the software using a IDE. While it is possible to work with any IDE that supports Gradle Projects, this guide will use IntelliJ as a basis.

You can download IntelliJ [here](https://www.jetbrains.com/idea/).

|  |  |
| --- | --- |
| Lightbulb | IntelliJ make use of Gradle and JavaFx plugins, and the project will need these plugins.  If you have disabled them, you can go to File > Settings > Plugins to enable them. |

2.2 Local Project Setup

To contribute to this project, you will need to work using a local copy of this project. To have a local copy of this project, you will need to:

1. Fork the repository from [here](https://github.com/CS2103JAN2018-W14-B4/main).
2. Clone the fork to your computer.
3. Open IntelliJ.

|  |  |
| --- | --- |
| Lightbulb | If you are not in the welcome screen, you can go to File > Close Project to close the exisiting project dialog. |

1. Set up the correct JDK version for Gradle.
   1. Click Configure > Project Defaults > Project Structure.
   2. Click New… and find the directory of the JDK.
2. Click Import Project.
3. Locate the build.gradle file, select it and click OK.
4. Click Open as Project.
5. Click OK to accept the default settings.
6. Open a console and run the command gradlew processResources (for Windows) or ./gradlew processResources (for MacOS/Linux).

|  |  |
| --- | --- |
| Lightbulb | This will generate all resources required by the application and tests, and finish with BUILD SUCCESSFUL message. |

2.3 Setup Verification

To ensure that you have set up the local copy of the project correctly, you will need to:

1. Run the seedu.address.MainApp and try the commands stated in the User Guide.
2. Run the tests and ensure that all the tests pass.

|  |  |
| --- | --- |
| Lightbulb | Section 6: Testing guides you on how to run tests for this project. |

2.4 Project Configurations

There are three project configurations that needs to be done to ensure that code and documentation quality are not compromised.

2.4.1 Coding Style Configurations

This project follows the [oss-generic coding standards](https://github.com/oss-generic/process/blob/master/docs/CodingStandards.adoc). IntelliJ uses a different package importing order, and to rectify:

1. Go to File > Settings... (for Windows/Linux) or IntelliJ IDEA > Preferences (for MacOS).
2. Select Editor > Code Style > Java.
3. Click on the Imports tab to set the order.
4. Set Class count to use import with '\*' and Names count to use static import with '\*' to 999.

|  |  |
| --- | --- |
| Lightbulb | This action prevents IntelliJ from contracting import statements. |

1. Set Import Layout order as:
   1. import static all other imports
   2. import java.\*
   3. import javax.\*
   4. import org.\*
   5. import com.\*
   6. import all other imports
2. Add a <blank line> between each import.

Alternatively, you can follow the [UsingCheckstyle documentation](https://github.com/CS2103JAN2018-W14-B4/main/blob/master/docs/UsingCheckstyle.adoc) to configure coding style.

2.4.2 Documentation Configurations

After forking the repository in [Section 2.2 Project Setup](#LocalProjectSetup), the links in the documentation will still lead to CS2103JAN2018-W14-B4/main repository. If you intend to develop this project as a separate product instead of contributing to the CS2103JAN2018-W14-B4/main repository, you should replace the variable repoURL in DeveloperGuide.adoc and UserGuide.adoc with the URL of your fork.

2.4.3 Continuous Integration (CI) Configurations

There are two CI configurations, Travis and AppVeyor, that you can set up.

Travis is an Unix-based software, while AppVeyor is a Windows-based software.

To set up Travis and AppVeyor for your fork, please refer to the [UsingTravis documentation](https://github.com/CS2103JAN2018-W14-B4/main/blob/master/docs/UsingTravis.adoc) and [UsingAppVeyor documentation](https://github.com/CS2103JAN2018-W14-B4/main/blob/master/docs/UsingAppVeyor.adoc) respectively.

|  |  |
| --- | --- |
| Lightbulb | Having both Travis CI and AppVeyor CI ensures your App works on both Unix-based platforms and Windows-based platforms. |

You should also set up coverage reporting for your team fork, if you have one. Please refer to [UsingCoveralls documentation](https://github.com/CS2103JAN2018-W14-B4/main/blob/master/docs/UsingCoveralls.adoc).

|  |  |
| --- | --- |
| Lightbulb | Coverage reporting could be useful for a team repository that hosts the final version of the project, but it is not that useful for your personal fork. |