Welcome to the Discord Bot Foundation’s documentation website!

Here you’ll discover all the information you need to harness the power of the Discord Bot Foundation (DBF) and easily develop fantastic Discord Bots. DBF is a revolutionary bootstrap framework that allows you to create Discord Bots faster than ever before.

The DBF, which is built on top of the Java Discord API (JDA), provides a stable foundation as well as a complete set of tools and resources to help you streamline your bot creation process. Whether you’re a seasoned developer or just starting out, our documentation page is your one-stop shop for everything Discord Bot Foundation has to offer.

You’ll find full instruction, comprehensive guidelines, and convenient reference materials covering every element of working with DBF on these pages. We’ve got you covered on everything from setting up your development environment to comprehending the complexities of the DBF-based architecture.

Whether you want to learn about the basics, delve into advanced features, or solve common problems, our documentation will be your valued companion. We believe that by providing a well-organized and comprehensive resource, we can help developers like you construct powerful and efficient Discord Bots that exceed expectations.

So buckle up, get your IDE ready, and join the Discord Bot Foundation on a coding journey. We can’t wait to see what you’ll create and accomplish with this amazing framework. Have fun coding!

If you have any concerns or require assistance, please contact our dedicated support team. Have fun exploring the Discord Bot Foundation documentation!

Getting started

You must be familiar with the Java programming language. If you haven’t already learned Java, we strongly advise you to do so before proceeding.

Before you start on the project you need the JDK and an IDE, IntelliJ IDEA or Eclipse is recommended.

To use **The Discord Bot Foundation**, you’ll need Java 17 and JitPack to import the library.

You can either choose to make your own project or copy the Discord Bot Template that has some examples on how to use the DBF.

You own projects can be accomplished by using either Maven or Gradle.  
Examples of how to import the project are provided below.

Maven:

<repositories>  
 <repository>  
 <id>jitpack.io</id>  
 <url>https://jitpack.io</url>  
 </repository>  
 </repositories>  
 <dependencies>  
 <dependency>  
 <groupId>com.github.Greazi-Times</groupId>  
 <artifactId>Discord\_Bot\_Foundation</artifactId>  
 <version>Tag</version>  
 </dependency>  
 </dependencies>

Grade:

allprojects {  
 repositories {  
 maven { url 'https://jitpack.io' }  
 }  
 }  
  
 dependencies {  
 implementation 'com.github.Greazi-Times:Discord\_Bot\_Foundation:Tag'  
 }

Creating the main instance

To be able to use the DBF we need to initialise the system, this needs to be done by extending the class SimpleBot to your main class. This will allow you to call some pre made methods that will startup the discord bot.

SimpleBot provides basic methods called:

/\*\*

\* Called way before everything starts

\* <p>

\* This method runs before everything else, meaning that nothing is loaded yet.

\* Settings, MySQL, JDA, etc. are not loaded yet!

\*/

protected void onPreLoad() {

}

/\*\*

\* Called the moment JDA is registerd

\*/

protected void onBotLoad() {

}

/\*\*

\* The main loading method, called when we are ready to load

\*/

protected abstract void onBotStart();

/\*\*

\* The main method called when we are about to shut down

\*/

protected void onBotStop() {

}

/\*\*

\* Invoked before settings were reloaded.

\*/

protected void onBotPreReload() {

}

/\*\*

\* Invoked after settings were reloaded.

\*/

protected void onBotReload() {

}

/\*\*

\* Register your commands, events, tasks and files here.

\* <p>

\* This is invoked when you do `/reload`

\*/

protected void onReloadableStart() {

// TODO: Add command that need to be registered here

}

These methods are called in the following order:

[Flowchart needs to be placed here]