

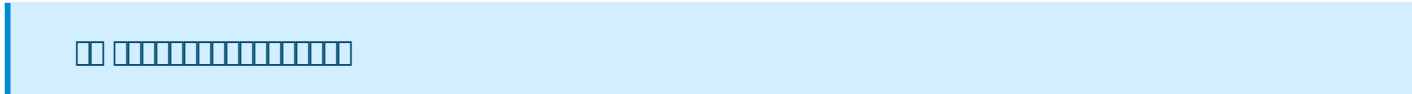
# Java

□□□□□

□□□□□□

- Java □□ - final □□□
- JDA □□ Discord Forum Channel □□ Thread □□
- Java □□ - Type Casting

# Java 面试题 - final 面试题



final 面试题 Java 面试题 Code 面试题

## 面试题 class 面试题

面试题 class 面试题

面试题 Compile error 面试题



面试题 Override 面试题

面试题 Compile error 面试题



面试题 Compile error 面试题

面试题 fianl



- 面试题 fianl 面试题 abstract 面试题
- 面试题 Reflection API 面试题 fianl 面试题



# JDA [ ] Discord Forum Channel [ ] Thread [ ]

## Code [ ]

```
public class Main extends ListenerAdapter {

    public static final String TOKEN = "<BOT_TOKEN_HERE>";
    public static final String SERVER_ID = "<SERVER_ID_HERE>";
    public static final String PROJECT_FORUM_ID = "<FORUM_ID_HERE>";

    public static void main(String[] args) throws InterruptedException {
        JDA jda = JDABuilder.createDefault(Main.TOKEN).build();

        jda.addEventListener(new Main());
    }

    @Override
    public void onReady(ReadyEvent event) {
        System.out.println("Bot is ready");
        JDA jda = event.getJDA();

        Guild server = jda.getGuildById(SERVER_ID);
        assert server != null;
        ForumChannel channel = server.getForumChannelById(PROJECT_FORUM_ID);
        assert channel != null;
        System.out.printf("Channel name: %s%n", channel.getName());
        List<ThreadChannel> threadChannels = channel.getThreadChannels();

        for(ThreadChannel thread : threadChannels){
            System.out.printf("Thread: %s%n", thread.getName());
        }
    }
}
```

```
// Bot
// jda.shutdown();
}
```



- JDA Guild (Discord Server) Guild ForumChannel
- getThreadChannels() ThreadChannel List
- List for



- TOKEN Channel ID (Hardcode)
- JDA (JDA.shutdown()) Gradle

# Java ☐☐ - Type Casting

[illegible]Casting ☐ ☐ ☐ ☐

```
class Animal {
    void speak(){
        System.out.println("Default speak");
    }
}
```

```
class Bird extends Animal {
    @Override
    void speak() {
        System.out.println("🐦 ");
    }
}
```

```
class Dog extends Animal {
    @Override
    void speak() {
        System.out.println("🐶");
    }
}
```

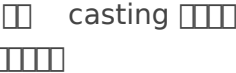


























```
public class main {  
    public static void main(String[] args) {  
        Animal animal = new Animal();  
        Bird bird = new Bird();  
        Dog dog = new Dog();  
  
        animal.speak();  
        bird.speak();  
        dog.speak();  
    }  
}
```

## // Upcasting and Downcasting

```
Animal animal2;
animal2 = bird; // Upcasting
animal2.speak(); // works normally
```

```
Bird bird2;
// bird2 = animal; // Down casing, compile error
// bird2 = (Bird)animal; // Compile pass
// bird2.speak(); // but throw ClassCastException error
```

```
Dog dog2;
// dog2 = bird; // Obviously not working for compiling
// dog2 = (Dog)bird; // Cannot casting to another child class, compile error.
}
}
```

- Bird  Animal  Upcasting  Upcasting 
-  animal casting  Animal  Bird   
 Compiler 
  -  Downcasting  Compiler 
-  (Bird)animal  Compiler  Compiler   
 Compiler  OK 
  -  ClassCastException  Runtime error   
 JVM 
-  Animal  class  
 Compiler   
 Compiler  ...