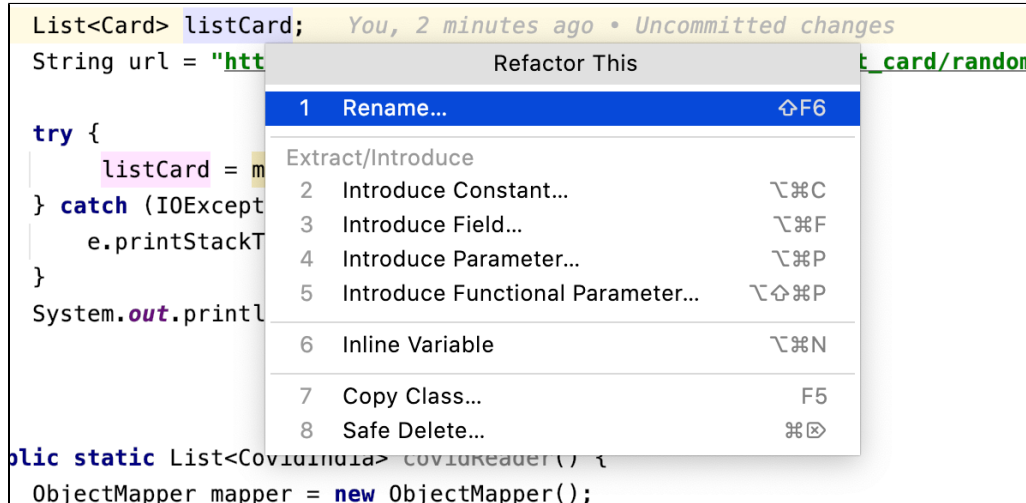


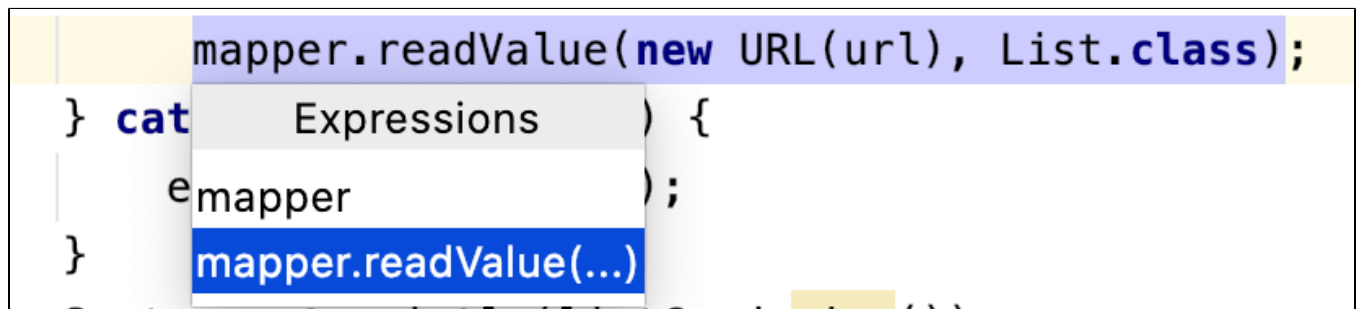
# IntelliJ Idea shortcuts

## Refractor

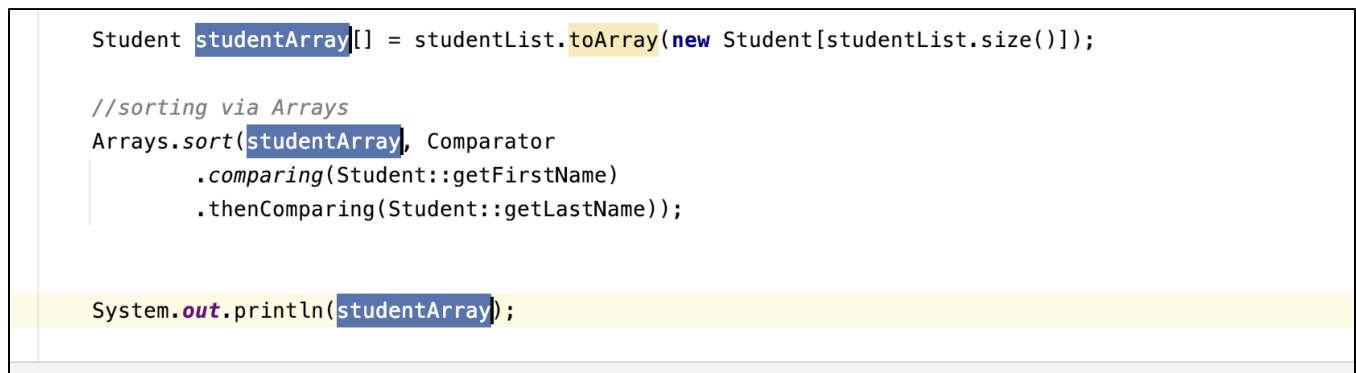
**Cmd Shift Opt T | Refactor This** - Opens up all the options below



**Cmd Opt V | Declare the variable**



**Ctrl Cmd G | Select All Occurrences**



## View

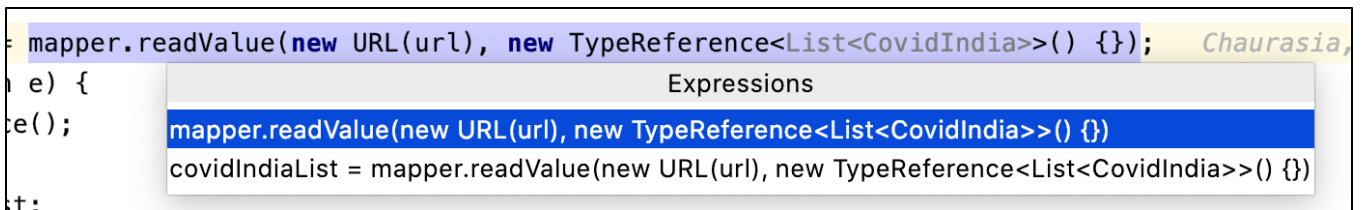
**Cmd Shift I | Quick Definition**, See definition of a symbol at caret



The screenshot shows an IDE with a code definition popup for the `readValue` method. The popup displays the following code:

```
@SuppressWarnings({ "unchecked" })
public <T> T readValue(URL src, TypeReference<T> valueTypeRef)
    throws IOException, StreamReadException, DatabindException
{
    _assertNotNull("src", src);
    return (T) _readMapAndClose(_jsonFactory.createParser(src), _typeFactory.c
}
```

**Ctrl Shift P | View Type Info**



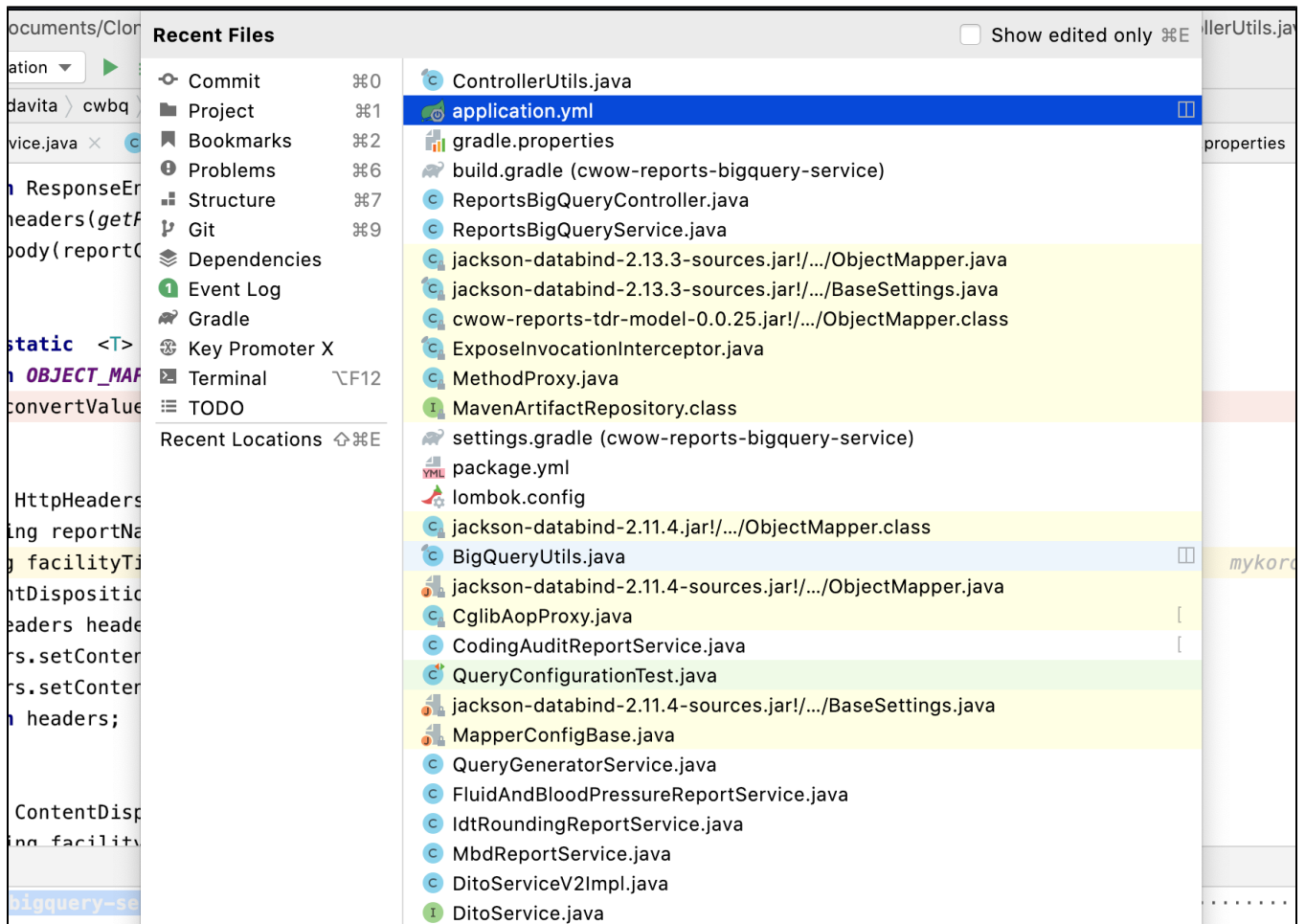
The screenshot shows an IDE with the 'Expressions' view for the `readValue` method. The view displays the following expressions:

```
mapper.readValue(new URL(url), new TypeReference<List<CovidIndia>>() {})
```



The screenshot shows an IDE with the 'List<CovidIndia>' type info. The type info is displayed as a tooltip over the `List<CovidIndia>` type in the code.

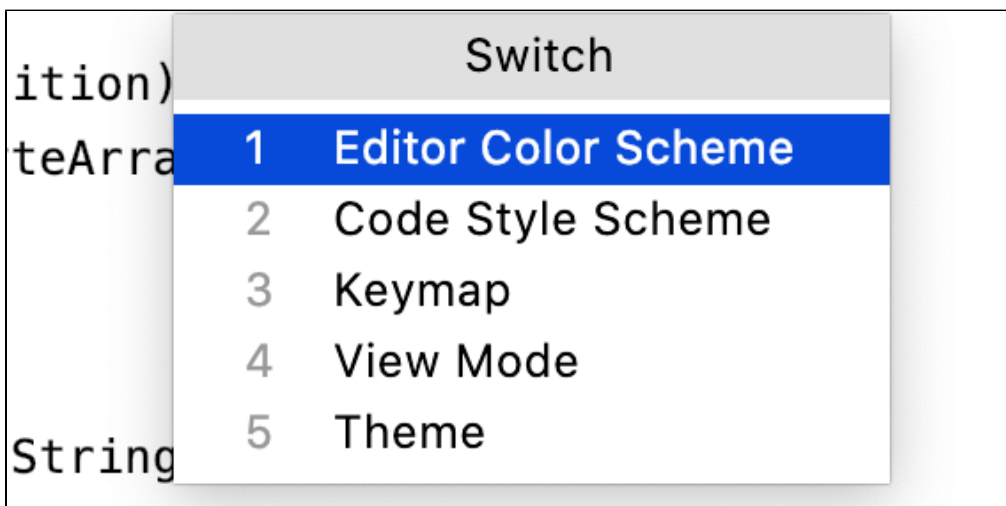
**Cmd + E | Move to the last location edited**, Show recently opened Files



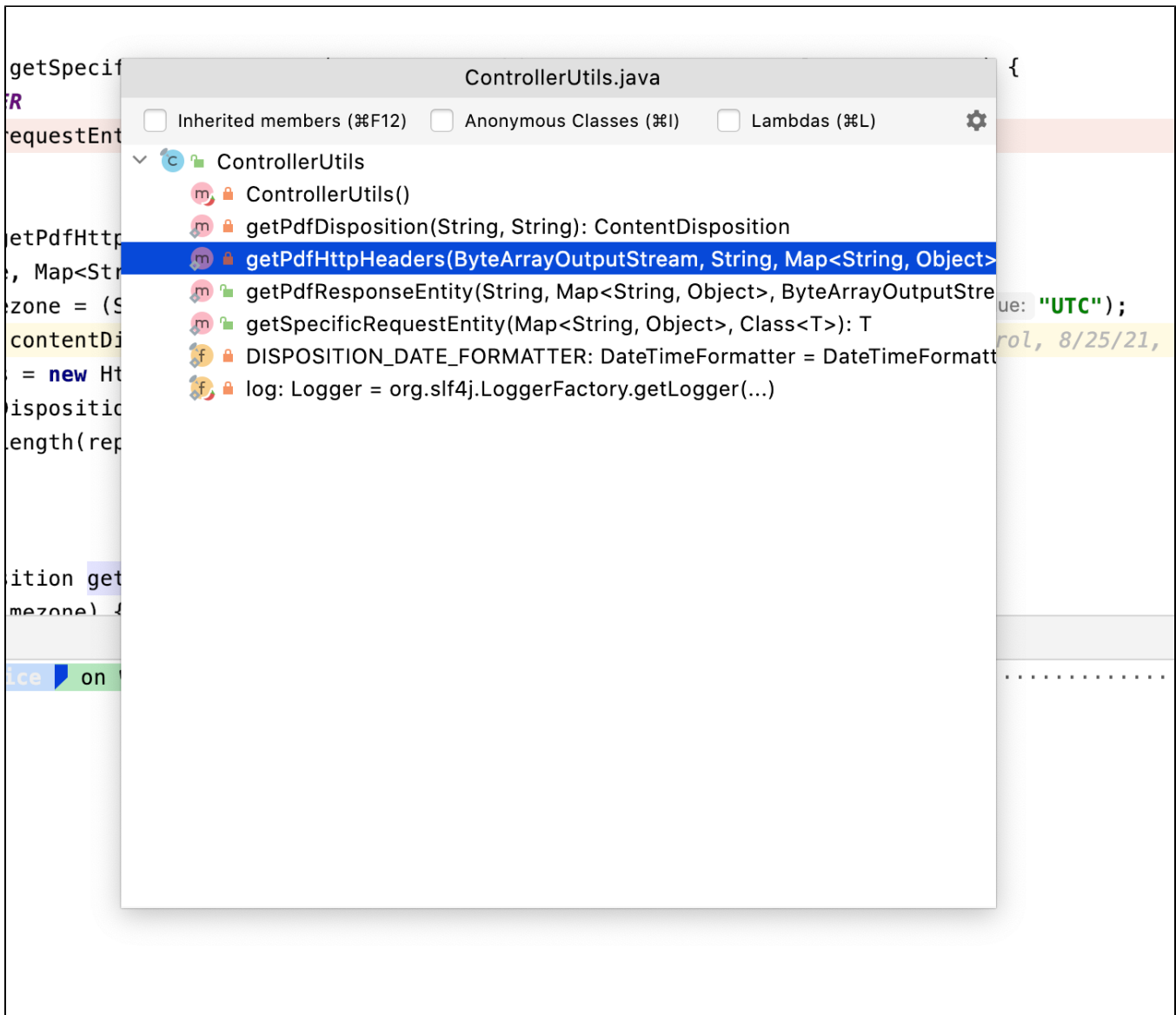
**Cmd Shift E | See edited code in recently opened Files, Recent Locations**



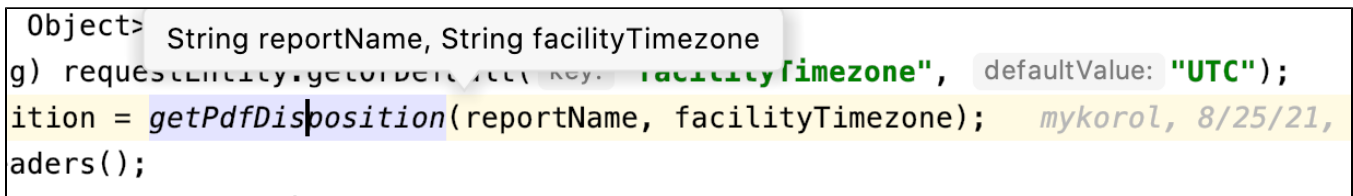
Ctrl ` Backtick | Quick Switch Scheme, enter presentation mode, etc



Cmd F12 | Display all methods in a file, on a pop up window | Cmd 7 Structure



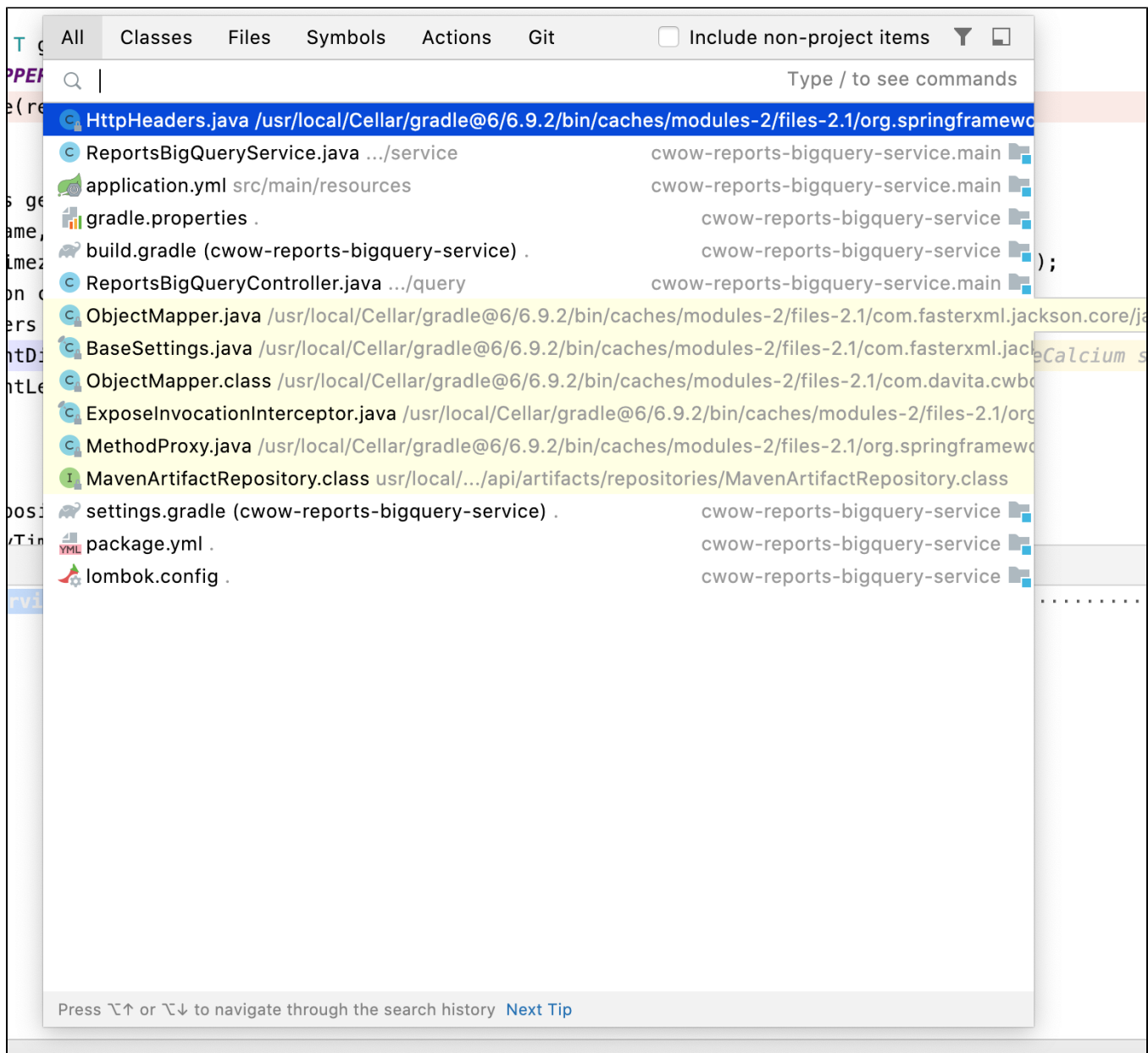
## Cmd P | View Parameter Information



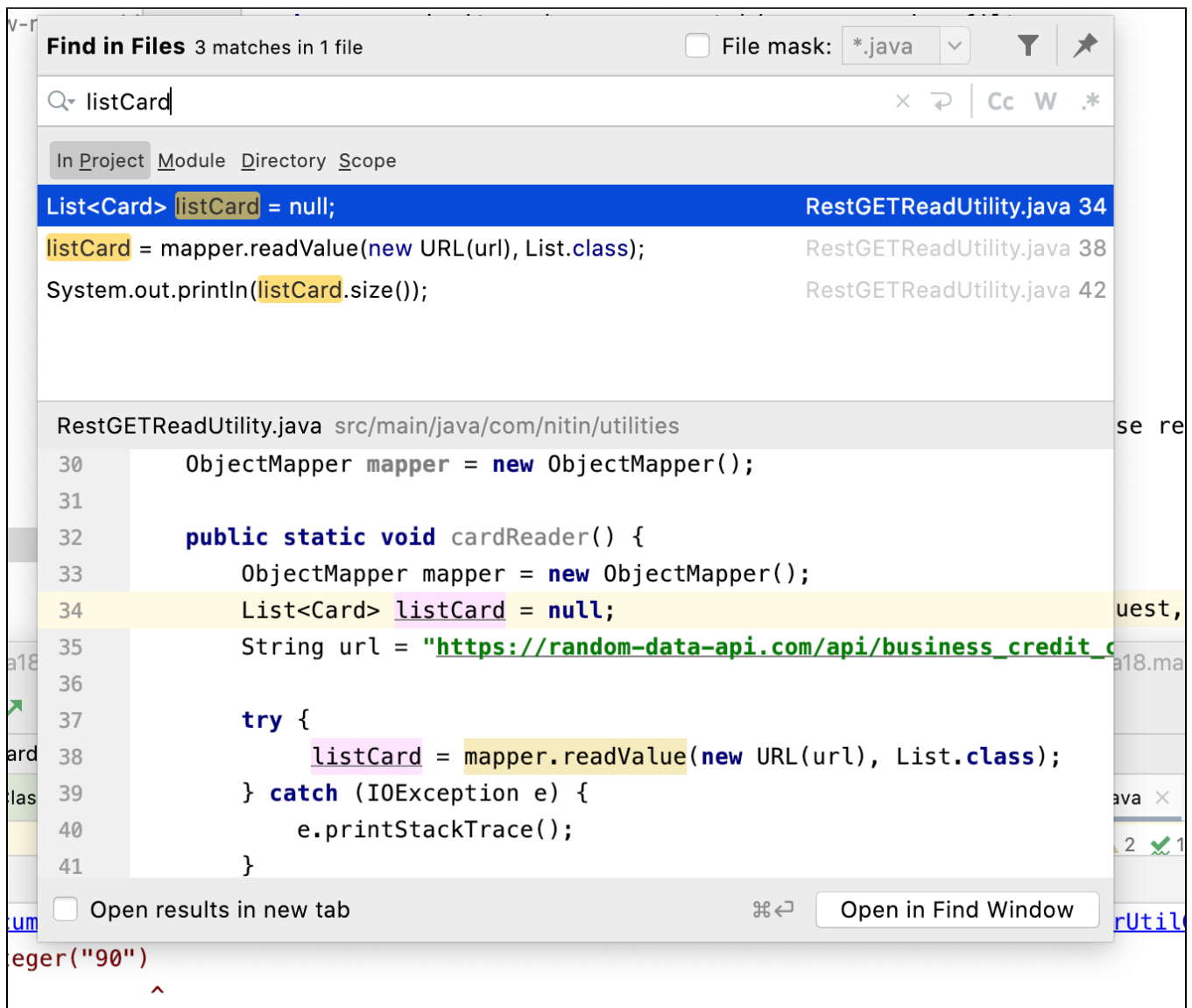
## Find, Search & Edit, Replace in Files

### Begin from

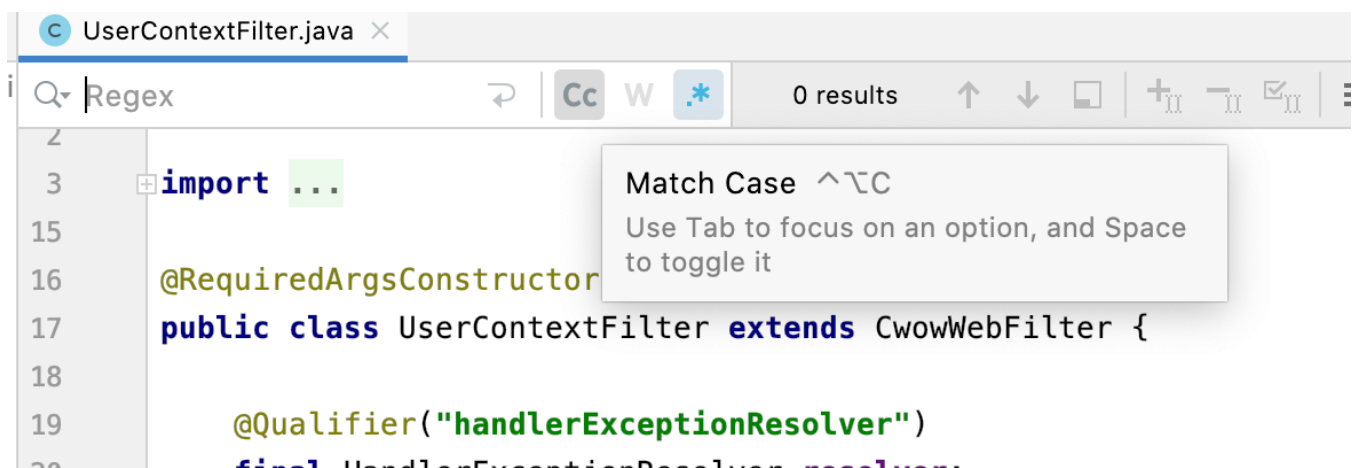
- **Shift Shift** | Find All | Common entry point for all menu
- **Cmd N** | Class
- **Cmd Shift N** -> File
- **Cmd Shift Opt N** -> Symbol



**^Ctrl Shift F | Find in Files**



### Cmd F | Find in current file



### Cmd Shift V | Paste from History...

*//Sorting via Streams*

List<Student> sortedStudentList = studentList.stream()

.sorted(Comparator

.comparing(Student::getFirstName)

.thenComparing(Student::getLastName))

.collect(Collectors.toList());

.collect(Collectors.toList()); //To Collect is IMPORTANT

Student studentArray[] = studentList.toArray(new Student[studentList.size()]);

*//sorting via Arrays*

Arrays.sort(studentArray, Comparator

.comparing(Student::getFirstName)

.thenComparing(Student::getLastName));

2 of 99

[chaurasia/Documents/ClonedCode/Java18/src/main/java/com/nitin/learning/UtilityClasses/Integerl](#)