File

File

In Java, the File class from **java.io** provides methods for managing files and directories. Key methods include **exists()** to check if a file or directory is present, **createNewFile()** to create a new file, and **delete()** to remove files or directories. It also includes **isDirectory()** and **isFile()** for type checking, **length()** for file size, and **getName()** and **getAbsolutePath()** for retrieving file details. Additionally, **renameTo()** allows renaming or moving files, and **listFiles()** lists contents of directories. System properties like **user.dir** (current working directory) and **user.home** (user's home directory) help construct paths relative to the application's or user's environment.

File

Handling file operations such as checking file existence, downloading, and uploading files is essential in test automation. You can check if a file exists using Java's **File** class with the **exists()** method. For downloads, configure browser settings, trigger the download, and verify the file's presence. File uploads are managed with Selenium’s **sendKeys()** method, sending the file path to an **input** element of **type file**. These techniques ensure files are correctly handled during automated tests.

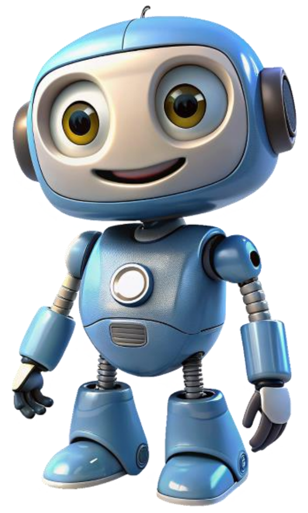
```
// Example 1: Check if a file exists
String filePath = "C:/path/to/your/file.txt";
File file = new File(filePath);
Assert.assertTrue("File does not exist.",file.exists());

// Example 2: Download a file
driver.get("http://example.com/download-file");
driver.findElement(By.id("downloadButton")).click();

// Wait for the file to be downloaded
File downloadedFile = new File(downloadFilePath + "/yourfile.txt");
Assert.assertTrue("File download failed.",downloadedFile.exists());

// Example 3: Upload a file
driver.get("http://example.com/upload-file");
driver.findElement(By.id("fileUpload")).sendKeys(filePath);
driver.findElement(By.id("uploadButton")).click();

// Verify the upload
String message = driver.findElement(By.id("successMessage")).getText();
Assert.assertTrue("File upload failed.",message.contains("Upload successful"));
```

talentify lab

Robot Class

Uploading files via the Robot class in Selenium is an alternative method for handling file uploads, especially in cases where the file upload dialog is **not directly accessible** through Selenium WebDriver. The Robot class in Java provides a way to simulate **keyboard** and **mouse** actions, which can be used to interact with **system dialogs**, such as the **file chooser** dialog that appears during file uploads.

```
// Click on the 'Choose File' button
driver.findElement(By.xpath("(//tr)[4]")).click();

// Set the file path to the clipboard
StringSelection ss = new StringSelection(path);
Toolkit.getDefaultToolkit().getSystemClipboard().setContents(ss, null);

// Create an instance of the Robot class
Robot robot = new Robot();

// Simulate pressing CTRL+V to paste the file path
robot.keyPress(KeyEvent.VK_CONTROL);
robot.keyPress(KeyEvent.VK_V);
robot.keyRelease(KeyEvent.VK_CONTROL);
robot.keyRelease(KeyEvent.VK_V);

// Simulate pressing ENTER to submit the file path
robot.keyPress(KeyEvent.VK_ENTER);
robot.keyRelease(KeyEvent.VK_ENTER);

// Click the submit button to upload the file
driver.findElement(By.xpath("//input[@type='submit']")).click();
```



talentify lab

talentify lab