**Методы сохранения загруженного файла или создания временного файла**

Храните его в любом доступном месте, кроме папки проекта IDE, также известной как папка развертывания сервера, по причинам, указанным в ответе на «Загруженное изображение», доступном только после обновления страницы:

1. Изменения в папке проекта IDE не сразу отражаются в рабочей папке сервера. В среде IDE есть своего рода фоновое задание, которое заботится о том, чтобы рабочая папка сервера синхронизировалась с последними обновлениями (в терминах IDE это называется «публикация»). Это основная причина проблемы, которую вы видите.

2. В реальном коде бывают ситуации, когда сохранение загруженных файлов в папке развертывания веб-приложения вообще не работает. Некоторые серверы (по умолчанию или по конфигурации) не разворачивают развернутый файл WAR в файловую систему локального диска, а полностью в памяти. Вы не можете создавать новые файлы в памяти без простого редактирования развернутого файла WAR и его повторного развертывания.

3. Даже когда сервер расширяет развернутый файл WAR в файловую систему локального диска, все вновь созданные файлы будут потеряны при повторном развертывании или даже простом перезапуске просто потому, что эти новые файлы не являются частью исходного файла WAR.

Для меня или кого-либо еще не имеет значения, где именно в файловой системе локального диска он будет сохранен, если вы никогда не используете метод getRealPath (). В любом случае использование этого метода вызывает тревогу.

Путь к месту хранения, в свою очередь, можно определить разными способами. Вы должны делать все сами.

Обратите внимание, что @MultipartConfig (location) не указывает конечное место назначения загрузки, а только место временного хранения файла размером , не превышающим размер установленный в файле *application.properties*.

Итак, путь к окончательному хранилищу можно определить одним из следующих **6** способов:

- **Hardcoded**:

File uploads = new File("/path/to/uploads");

- **Переменная среды** через SET UPLOAD\_LOCATION = / path / to / uploads:

File uploads = new File(System.getenv("UPLOAD\_LOCATION"));

- **Аргумент VM** при запуске сервера через -Dupload.location = "/ path / to / uploads":

File uploads = new File(System.getProperty("upload.location"));

- **Файл \*.properties** записать как upload.location = / path / to / uploads:

File uploads = new File(properties.getProperty("upload.location"));

- **web.xml** <context-param> с именем “upload.location” и значением “/ path / to / uploads”:

File uploads = new File(getServletContext().getInitParameter("upload.location"));

- Если есть, используйте **местоположени**е, указанное сервером, например в JBoss AS / WildFly:

File uploads = new File(System.getProperty("jboss.server.data.dir"), "uploads");

Сохраняем файл следующим образом:

File file = new File(uploads, "somefilename.ext"); //тут “uploads”, это из строк выше

try (InputStream input = part.getInputStream()) {

Files.copy(input, file.toPath());

}

- Или, если вы хотите автоматически сгенерировать уникальное имя файла, чтобы пользователи не перезаписывали существующие файлы с совпадающим именем:

File file = File.createTempFile("somefilename-", ".ext", uploads);

try (InputStream input = part.getInputStream()) {

Files.copy(input, file.toPath(), StandardCopyOption.REPLACE\_EXISTING);

}

Как получить часть в JSP / Servlet, ответ на вопрос Как загрузить файлы на сервер с помощью JSP / Servlet? и как получить часть в JSF, ответ дан в Как загрузить файл с помощью JSF 2.2 <h: inputFile>? Где сохраненный файл?

*Примечание:* не используйте Part # write (), поскольку он интерпретирует путь относительно временного хранилища, определенного в @MultipartConfig (location).