1. Назначение компонента

Компонент предназначен для обработки запросов стандартного вида через определенных обработчиков. На основании переданных данных в JSON, используется определенный обработчик, выполняя запросы на получение данных, создание и редактирование данных.

Основными назначениями компонента является:

- авторизация;

- запросы к базе данных;

- работы с файловой системой (через базу данных или, непосредственно, файловую систему).

Работа с компонентом осуществляется с помощью стандартизированных JSON запросов.

1. Лицензия

Лицензия проекта - Eustrosoft BALES. Её описание можно найти в корне проекта.

1. Какие сторонние компоненты (библиотеки, фреймворки, компиляторы, препроцессоры) используются, для чего.

Компиляторы:

- Java 8 (OpenJDK 8 [1.8.301]).

Библотеки:

- javax.servlet-api 4.0.1;

- jsp-api 2.0 (для тестирования);

- org.postgresql (42.5.1);

- com.google.code.gson 2.10.1 (формирование ответов);

- org.projectlombok - Lombok 1.18.24;

- commons-io 2.11.0;

- org.eustrosoft.DBPool 0.0.1;

- org.eustrosoft.SessionCookie 0.0.1;

Сборщик проекта:

- apache maven 3.6.2

- org.apache.maven.plugins (maven-war-plugin, maven-compiler-plugin, projectlombok plugin);

Используемый сервер:

- apache tomcat 9.x.

1. Как собрать (из командной строки), как установить в эксплуатацию

Для начала необходимо скачать библиотеки, которые не находятся в хранилищах maven.

Библиотеки находятся на сайте:

- <http://ftp.eustrosoft.org/pub/eustrosoft.org/pkg/ConcepTIS/>

Необходимо скачать два JAR файла:

- qDBPool;

- qSessionCookie.

После этого их необходимо установить в папку m2 для пользователя, который будет собрать проект. Для этого описан скрипт, находящийся в корне проекта в папке «scripts». Скрипт называется “installMvn”, и он включает в себя подстановки, которые необходимо заполнить. Руководствоваться по именованию и версиям необходимо, учитывая данные в файле mvn основного проекта.

Также необходимо не забыть поменять параметры в файле cms.properties и logging.properties в соответствие с предпочтениями

Данные файлы включают в себя конфигурацию логов и параметров для кмс системы.

Версия, имя и пути должны соответствовать тем, что находятся в мавен файле проекта.

Далее необходимо собрать основной проект. Базовая сборка – в вар файл, который необходимо поставить в папку томката для веб приложений.

1. Структура дерева исходных текстов, где что лежит, сколько строк

Слева – количество строк, справа – относительный путь в проекте.

31 ./src/main/java/com/eustrosoft/core/AliveServlet.java

61 ./src/main/java/com/eustrosoft/core/Constants.java

40 ./src/main/java/com/eustrosoft/core/context/DBPoolContext.java

15 ./src/main/java/com/eustrosoft/core/context/EustrosoftUser.java

15 ./src/main/java/com/eustrosoft/core/context/StorageContext.java

46 ./src/main/java/com/eustrosoft/core/context/User.java

67 ./src/main/java/com/eustrosoft/core/context/UsersContext.java

144 ./src/main/java/com/eustrosoft/core/context/UserStorage.java

33 ./src/main/java/com/eustrosoft/core/filter/CorsAllowFilter.java

92 ./src/main/java/com/eustrosoft/core/filter/RequestLoggingFilter.java

26 ./src/main/java/com/eustrosoft/core/handlers/BasicAuthHandler.java

272 ./src/main/java/com/eustrosoft/core/handlers/cms/CMSHandler.java

128 ./src/main/java/com/eustrosoft/core/handlers/cms/CMSRequestBlock.java

79 ./src/main/java/com/eustrosoft/core/handlers/cms/CMSResponseBlock.java

22 ./src/main/java/com/eustrosoft/core/handlers/cms/DownloadFileDetails.java

11 ./src/main/java/com/eustrosoft/core/handlers/cms/FileDownloadMap.java

19 ./src/main/java/com/eustrosoft/core/handlers/cms/FileTicket.java

64 ./src/main/java/com/eustrosoft/core/handlers/ExceptionBlock.java

131 ./src/main/java/com/eustrosoft/core/handlers/file/BytesChunkFileHandler.java

28 ./src/main/java/com/eustrosoft/core/handlers/file/BytesChunkFileRequestBlock.java

118 ./src/main/java/com/eustrosoft/core/handlers/file/ChunkFileHandler.java

55 ./src/main/java/com/eustrosoft/core/handlers/file/ChunkFileRequestBlock.java

53 ./src/main/java/com/eustrosoft/core/handlers/file/FileHandler.java

150 ./src/main/java/com/eustrosoft/core/handlers/file/FileRequestBlock.java

51 ./src/main/java/com/eustrosoft/core/handlers/file/FileResponseBlock.java

133 ./src/main/java/com/eustrosoft/core/handlers/file/HexFileHandler.java

45 ./src/main/java/com/eustrosoft/core/handlers/file/HexFileRequestBlock.java

14 ./src/main/java/com/eustrosoft/core/handlers/Handler.java

129 ./src/main/java/com/eustrosoft/core/handlers/login/LoginHandler.java

75 ./src/main/java/com/eustrosoft/core/handlers/login/LoginRequestBlock.java

64 ./src/main/java/com/eustrosoft/core/handlers/login/LoginResponseBlock.java

53 ./src/main/java/com/eustrosoft/core/handlers/ping/PingHandler.java

55 ./src/main/java/com/eustrosoft/core/handlers/ping/PingRequestBlock.java

91 ./src/main/java/com/eustrosoft/core/handlers/ping/PingResponseBlock.java

30 ./src/main/java/com/eustrosoft/core/handlers/requests/BasicRequest.java

16 ./src/main/java/com/eustrosoft/core/handlers/requests/MethodConstants.java

22 ./src/main/java/com/eustrosoft/core/handlers/requests/QTisRequestObject.java

15 ./src/main/java/com/eustrosoft/core/handlers/requests/Request.java

20 ./src/main/java/com/eustrosoft/core/handlers/requests/RequestBlock.java

52 ./src/main/java/com/eustrosoft/core/handlers/requests/RequestObject.java

21 ./src/main/java/com/eustrosoft/core/handlers/responses/BasicResponse.java

79 ./src/main/java/com/eustrosoft/core/handlers/responses/QTisResponse.java

19 ./src/main/java/com/eustrosoft/core/handlers/responses/Response.java

30 ./src/main/java/com/eustrosoft/core/handlers/responses/ResponseBlock.java

11 ./src/main/java/com/eustrosoft/core/handlers/responses/ResponseLang.java

76 ./src/main/java/com/eustrosoft/core/handlers/sql/DBConnector.java

78 ./src/main/java/com/eustrosoft/core/handlers/sql/DBWrapper.java

58 ./src/main/java/com/eustrosoft/core/handlers/sql/model/ResultSetAnswer.java

66 ./src/main/java/com/eustrosoft/core/handlers/sql/SQLHandler.java

72 ./src/main/java/com/eustrosoft/core/handlers/sql/SQLRequestBlock.java

128 ./src/main/java/com/eustrosoft/core/handlers/sql/SQLResponseBlock.java

45 ./src/main/java/com/eustrosoft/core/handlers/sql/utils/ResultSetUtils.java

459 ./src/main/java/com/eustrosoft/core/HttpRequestDispatcher.java

93 ./src/main/java/com/eustrosoft/core/providers/DataSourceProvider.java

47 ./src/main/java/com/eustrosoft/core/providers/SessionProvider.java

36 ./src/main/java/com/eustrosoft/core/tools/ColorTextUtil.java

70 ./src/main/java/com/eustrosoft/core/tools/FileDownloadService.java

122 ./src/main/java/com/eustrosoft/core/tools/FileUtils.java

77 ./src/main/java/com/eustrosoft/core/tools/Json.java

29 ./src/main/java/com/eustrosoft/core/tools/LogFormatter.java

19 ./src/main/java/com/eustrosoft/core/tools/PropertiesConstants.java

612 ./src/main/java/com/eustrosoft/core/tools/QJson.java

34 ./src/main/java/com/eustrosoft/core/tools/WebParams.java

48 ./src/main/java/com/eustrosoft/core/tools/ZipService.java

13 ./src/main/java/com/eustrosoft/datasource/exception/CMSException.java

65 ./src/main/java/com/eustrosoft/datasource/sources/CMSDataSource.java

11 ./src/main/java/com/eustrosoft/datasource/sources/Constants.java

23 ./src/main/java/com/eustrosoft/datasource/sources/FileDetails.java

31 ./src/main/java/com/eustrosoft/datasource/sources/HexFileParams.java

27 ./src/main/java/com/eustrosoft/datasource/sources/HexFileResult.java

34 ./src/main/java/com/eustrosoft/datasource/sources/model/CMSDirectory.java

36 ./src/main/java/com/eustrosoft/datasource/sources/model/CMSFile.java

36 ./src/main/java/com/eustrosoft/datasource/sources/model/CMSGeneralObject.java

56 ./src/main/java/com/eustrosoft/datasource/sources/model/CMSLink.java

35 ./src/main/java/com/eustrosoft/datasource/sources/model/CMSObject.java

34 ./src/main/java/com/eustrosoft/datasource/sources/model/CMSType.java

22 ./src/main/java/com/eustrosoft/datasource/sources/parameters/CMSObjectUpdateParameters.java

40 ./src/main/java/com/eustrosoft/datasource/sources/PropsContainer.java

34 ./src/main/java/com/eustrosoft/datasource/sources/Source.java

44 ./src/main/java/com/eustrosoft/dbdatasource/constants/DBConstants.java

30 ./src/main/java/com/eustrosoft/dbdatasource/constants/SQLConstants.java

493 ./src/main/java/com/eustrosoft/dbdatasource/core/DBDataSource.java

490 ./src/main/java/com/eustrosoft/dbdatasource/core/DBFunctions.java

253 ./src/main/java/com/eustrosoft/dbdatasource/core/DBStatements.java

81 ./src/main/java/com/eustrosoft/dbdatasource/core/ExecStatus.java

21 ./src/main/java/com/eustrosoft/dbdatasource/core/model/DBObject.java

68 ./src/main/java/com/eustrosoft/dbdatasource/core/model/FDir.java

115 ./src/main/java/com/eustrosoft/dbdatasource/core/model/FFile.java

9 ./src/main/java/com/eustrosoft/dbdatasource/core/model/JsonFormat.java

9 ./src/main/java/com/eustrosoft/dbdatasource/core/model/ResultSetConverter.java

5 ./src/main/java/com/eustrosoft/dbdatasource/core/model/Updatable.java

161 ./src/main/java/com/eustrosoft/dbdatasource/queries/Query.java

38 ./src/main/java/com/eustrosoft/dbdatasource/ranges/FileType.java

73 ./src/main/java/com/eustrosoft/dbdatasource/util/ResultSetUtils.java

30 ./src/main/java/com/eustrosoft/filedatasource/constants/FileConstants.java

35 ./src/main/java/com/eustrosoft/filedatasource/constants/Messages.java

417 ./src/main/java/com/eustrosoft/filedatasource/FileCMSDataSource.java

42 ./src/main/java/com/eustrosoft/filedatasource/util/FileUtils.java

316 ./src/main/java/com/eustrosoft/tools/CMSDownloader.java

16 ./src/main/java/com/eustrosoft/tools/Constants.java

43 ./src/main/java/com/eustrosoft/tools/DateFormatter.java

67 ./src/main/java/com/eustrosoft/tools/HexFileConverter.java

1. Архитектура проекта, что за что отвечает

Для начала в проекте существуют константы и утильные классы, которые работают независимо от каких-либо компонентов. Этими классами являются все классы в папках утил и файлы с названием «Константы» в имени файла.

Двумя входными точками в проект являются два сервлета. Один из них показывает доступность сервлета – AliveServlet. Второй – непосредственно, обработчик – HttpRequestDispatcher (com.eustrosoft.core).

Маппинг сервлетов осуществляется с помощью аннотаций.

Архитектура обработки запросов построена следующим образом:

1. запрос приходит в сервлет и разбирается на составные части.
2. Определяется, какой обработчик будет обрабатывать тот или иной запрос. Потому что запросов внутренних для одного хттп запроса может быть несколько.
3. Запрос обрабатывается обработчиком и он присылает ответ в стандартном формате.

В проекте существует несколько интерфейсов: запрос, ответ, обработчик. Это базовые интерфейсы, которые обязан реализовывать каждый тип запроса, ответа или обработчик.

Другими словами – это контракт, который гарантирует, что обработка запроса пойдет в соответствие с планом. Интерфейсы запроса и ответа реализуют абстрактные классы BasicRequest и BasicResponse – они облегчают написание объектов запроса для определенных обработчиков. Такой подход выбран в связи с разными требованиями к каждому определенному обработчику, которым необходимы разные параметры с разной компоновкой. Таким образом – запросы и ответы – отдельные сущности, которыми можно манипулировать и фильтровать, если это будет необходимо. В данном варианте для обработчика приходит интерфейс запроса, он кастится к типу запроса, который необходим обработчику, и он работает с ним в своих интересах. Далее – получаем стандартный ответ, который обрабатываем через библиотеку GSON для отправки ответа.

1. Протоколы взаимодействия между компонентами

Взаимодействие между Front-End и Back-End происходят с помощью стандартизированного json формата.

Стандарт описан в файле в проекте Documentation.

Перечисление API для работы с сервисом:

Запросы с авторизацией:

- POST /api/dispatch – необходим для обработки запросов в стандартном формате;

- GET /api/download – необходим для скачивания файлов из подсистемы CMS. В качестве параметра принимает id (ID – объекта для скачивания), path (полный путь до файла, который необходимо скачать), contentType (тип контента, который будет указан в ответе).

Запросы без авторизации:

- GET /alive – Получить состояние сервиса. 200 со значением “Alive!” или 500.

1. Направления доработки и развития

В идеальном варианте необходимо, чтобы процесс написания интерфейсов для запросов был минимален, а процесс перенаправления запросов был реализован не через написанную логику, а через определения аннотаций, экономя время для перенаправлений запросов в необходимый обработчик.

Также, необходимо обрабатывать моменты с ошибками при разборе запроса, ведь он может прийти некорректный. Создать собственный класс исключений. Провести рефактор классов и пакетов, удалить не использующийся или устаревший код.

Провести рефактор сервлетов, оставить в них минимальное количество кода, всю обработку предоставить другим классам, занимающимися разбором запросов и их роутингом в определенные обработчики.

9. План рефакторинга

10. Выводы. Что было сделано правильно, что не правильно

Правильно:  
1. Контракт обработки запросов. Каждый запрос и ответ прозрачны – они имеют собственные классы и их легко отлаживать. Всегда понятно, на каком этапе произошла ошибка и где смотреть.

2. Классы, хранящие в себе состояние чего-либо. Например – соединение с бд. Данными классами можно пользоваться по всему проекту. Это облегчает их использование там, где это необходимо, без излишних действий по логину пользователя, проверки куки.

Неправильно:  
1. Лишняя библиотека json. Необходимо оставить одну.

2. Интерфейсы DataSource (CMS part) – очень перенасыщены методами. Их необходимо разбить на отдельные интерфейсы или пересмотреть.

3. Сервлеты также перенасыщены кодом. Они должны оставаться лаконичными и читаемыми, вся обработка запросов должна уйти в другие классы со своей архитектурой.

4. В пакете для обработчиков – оставить минимально необходимые пакеты и классы, все остальное – вынести в другой пакет или библиотеку.

5. Конфигурации присутствуют что в конфигурационных файлах, что в web.xml. Необходимо определиться с их назначением, локацией и выбрать что-то одно.

6. Есть типы запросов, которые могут не обработаться правильно. В применении с фронт-ендом этого может быть незаметно, но при дебаге запросами на сервер этого добиться просто, достаточно прислать запрос не соответствующий протоколу. Для этого необходимо

1) почистить сервлеты;

2) создать классы «разбор»а запросов;

3) определить логику при нарушении протокола.

7. Весь пакер обработки sql запросов нужно пересмотреть, использовать актуальные средства исполнения запросов и самые надежные.

8. Также, в пакете исполнения запросов могут появиться глюки. Если при создании файла вывалится ошибка – может быть заблокирована папка и не не закоммичена. Это ухудшит юзер-экспириенс.

9. Скачивание в 1 мегабайт может быть достаточно большим объемом, но следует и учитывать заголовки запроса, остальные данные, проходящие через сетевой канал и выбрать актуальный размер для чанков.