

## Содержание

<b>1</b>	<b>Вступление .....</b>	<b>2</b>
<b>2</b>	<b>Java.....</b>	<b>3</b>
<b>3</b>	<b>WildFly.....</b>	<b>3</b>
3.1	Ошибка при запуске .....	3
3.2	Добавление пользователя.....	5
<b>4</b>	<b>Среда разработки .....</b>	<b>6</b>
<b>5</b>	<b>База данных.....</b>	<b>7</b>
<b>6</b>	<b>Создание проекта в IntelliJ Idea.....</b>	<b>11</b>
6.1	Создание модулей.....	11
6.2	Создание артефакта .....	17
6.3	Создание конфигурации запуска .....	19
6.4	Добавление библиотеки.....	24
6.5	Запуск.....	26
<b>7</b>	<b>Структура проекта .....</b>	<b>26</b>
7.1	Контроллеры.....	28
7.2	Модели.....	29
<b>8</b>	<b>Билд .....</b>	<b>31</b>
<b>9</b>	<b>Работа с WildFly .....</b>	<b>32</b>
9.1	Деплой на сервер приложений.....	32
9.2	Перезагрузка приложения.....	34

## 1 Вступление

Репозиторий инсталляционных пакетов разработан на языке Java с использованием Spring framework 4. Руководство ориентировано на разработку в ОС Windows.

Реализованные функции:

Для пользователей:

- Сквозная аутентификация через Active Directory;
- Регистрация заявок на добавление инсталляционных пакетов;
- Сохранений заявок в базе данных;
- Загрузка файлов на сервер;
- Просмотр каталога ПО;
- Просмотр информации о файле;
- Выгрузка файлов с сервера.

Для модератора:

- Сквозная аутентификация через Active Directory;
- Просмотр каталога ПО;
- Просмотр информации о файле;
- Загрузка файлов на сервер;
- Выгрузка файлов с сервера;
- Заполнение информации о файлах; тип информации будет уточнен в процессе разработки;
- Редактирование информации о файлах;
- Сохранение информации о файлах в базе данных;
- Просмотр списка файлов;
- Создание новых категорий для группировки файлов;
- Сохранение категорий в базе данных;
- Удаление категорий;

- Привязка файлов к категориям;
- Просмотр файлов, информация о которых еще не заполнена;
- Просмотр информации о пользователях; тип информации будет уточнен в процессе разработки.

Для администратора системы:









- Все возможности модератора;
- Изменить настройки;
- Удаление файлов;
- Выполнение скриптов для экспорта файлов в SCCM.

## 2 Java

Для начала разработки необходимо установить Java JDK, которую мы можем скачать с официального сайта Oracle.

## 3 WildFly

Для запуска приложения нужен сервер приложений WildFly. Скачать можно с официального сайта.

Downloads					
Version	Date	Description	License	Size	Format
10.0.0.Final	2016-01-29	Java EE7 Full & Web Distribution	LGPL	132 MB	 <a href="#">ZIP</a>
				131 MB	 <a href="#">TGZ</a>
		Servlet-Only Distribution	LGPL	28 MB	 <a href="#">ZIP</a>
				28 MB	 <a href="#">TGZ</a>
		Application Server Source Code	LGPL	25 MB	 <a href="#">ZIP</a>
				15 MB	 <a href="#">TGZ</a>
		Quick Start Source Code	AL		 <a href="#">Source</a>
		Release Notes			 <a href="#">Notes</a>

Скачиваем и распаковываем архив куда-нибудь.

Для запуска сервера необходимо запустить bin/standalone.bat.

### 3.1 Ошибка при запуске

Во время запуска на Windows может возникнуть ошибка вида:

```
10:50:41:201 ERROR [org.jboss.msc.service.fail] (MSC service thread 1-3) MSC000001: Failed to start service jboss.server.management.controller.management-http: org.jboss.msc.service.StartException in service jboss.server.management.controller.management-http: WFLYSRV0083: Failed to start the http-interface service
    at org.jboss.as.server.mgmt.UndertowHttpManagementService.start(UndertowHttpManagementService.java:221)
    at org.jboss.msc.service.ServiceControllerImpl$StartTask.startService(ServiceControllerImpl.java:1948)
    at org.jboss.msc.service.ServiceControllerImpl$StartTask.run(ServiceControllerImpl.java:1881)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
    at java.lang.Thread.run(Unknown Source)
Caused by: java.lang.RuntimeException: java.net.BindException: Address already in use: bind
    at org.jboss.as.domain.http.server.ManagementHttpServer.start(ManagementHttpServer.java:157)
    at org.jboss.as.server.mgmt.UndertowHttpManagementService.start(UndertowHttpManagementService.java:237)
    ... 5 more
Caused by: java.net.BindException: Address already in use: bind
    at sun.nio.ch.Net.bind(Native Method)
    at sun.nio.ch.Net.bind(Unknown Source)
    at sun.nio.ch.ServerSocketChannelImpl.bind(Unknown Source)
    at sun.nio.ch.ServerSocketAdaptor.bind(Unknown Source)
    at org.xmlio.nio.NioNioWorker.createTcpConnectionServer(NioNioWorker.java:192)
    at org.xmlio.NioNioWorker.createStreamConnectionServer(NioNioWorker.java:243)
    at org.jboss.as.domain.http.server.ManagementHttpServer.start(ManagementHttpServer.java:144)
    ... 6 more
```

Это значит, что требуемые для запуска порты заняты. Для того, чтобы изменить порты открываем standalone/configuration/standalone.xml, ищем внизу файла строки:

```
<socket-binding-group name="standard-sockets" default-interface="public" port-
offset="${jboss.socket.binding.port-offset:0}">

    <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:9990}"/>

    <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:9993}"/>

    <socket-binding name="ajp" port="${jboss.ajp.port:8009}"/>

    <socket-binding name="http" port="${jboss.http.port:8080}"/>

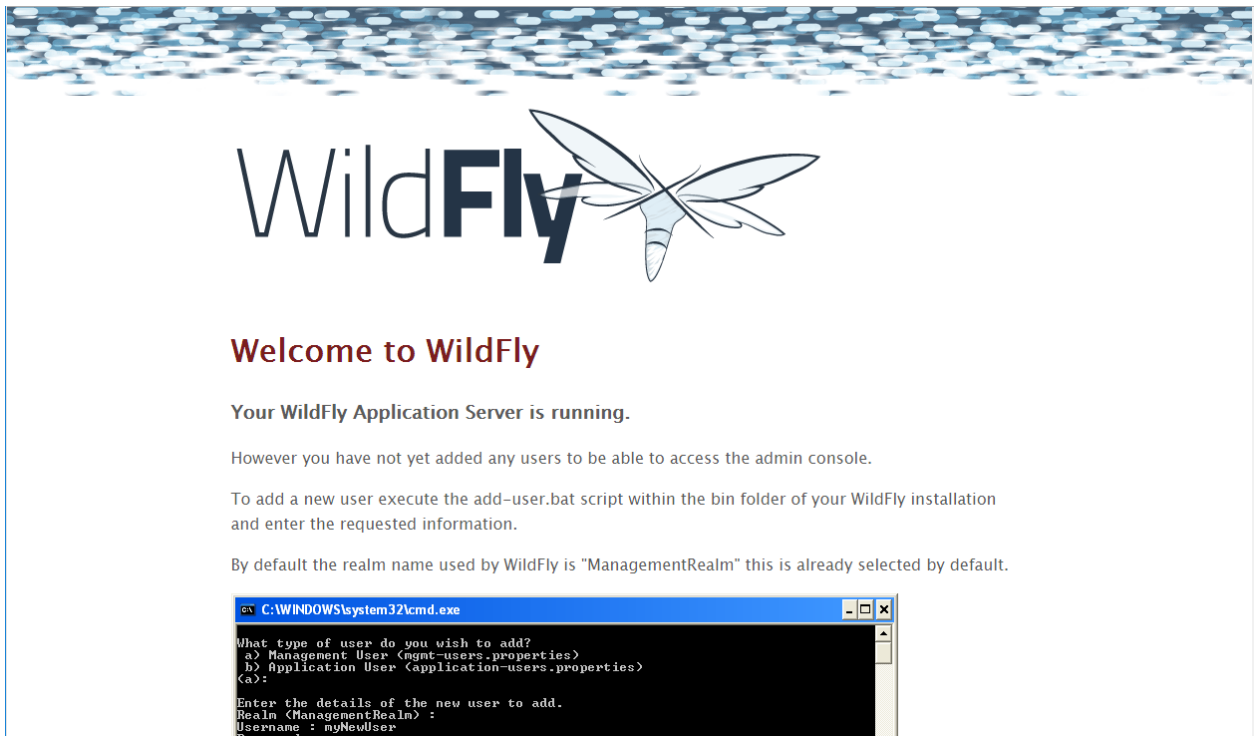
    <socket-binding name="https" port="${jboss.https.port:8443}"/>

    <socket-binding name="txn-recovery-environment" port="4712"/>

    <socket-binding name="txn-status-manager" port="4713"/>

    <outbound-socket-binding name="mail-smtp">
        <remote-destination host="localhost" port="25"/>
    </outbound-socket-binding>
</socket-binding-group>
```

Необходимо изменить порты в `${jboss.management.http.port:9990}` и `${jboss.management.https.port:9993}` на другие. В данном случае, порт 9990 занят драйверами NVidia, поэтому меняем 9990 на 9991, сохраняем и запускаем сервер еще раз. Если все прошло без ошибок, то открываем <http://127.0.0.1:9991/>, и мы должны увидеть стартовую страницу:



### 3.2 Добавление пользователя

На стартовой странице написано, что нужно добавить пользователя. Для этого, запускаем `/bin/add-user.bat`.

```
JAVA_HOME is not set. Unexpected results may occur.
Set JAVA_HOME to the directory of your local JDK to avoid this message.

What type of user do you wish to add?
a) Management User (mgmt-users.properties)
b) Application User (application-users.properties)
(a):
```

Выбираем a) Management user и вводим все что у нас просят:

```

JAVA_HOME is not set. Unexpected results may occur.
Set JAVA_HOME to the directory of your local JDK to avoid this message.

What type of user do you wish to add?
  a) Management User (mgmt-users.properties)
  b) Application User (application-users.properties)
(a): a

Enter the details of the new user to add.
Using realm 'ManagementRealm' as discovered from the existing property files.
Username : admin
The username 'admin' is easy to guess
Are you sure you want to add user 'admin' yes/no? yes
Password recommendations are listed below. To modify these restrictions edit the add-user.properties configuration file.

- The password should be different from the username
- The password should not be one of the following restricted values {root, admin, administrator}
- The password should contain at least 8 characters, 1 alphabetic character(s), 1 digit(s), 1 non-alphanumeric symbol(s)
)
Password :
WFLYDM0098: The password should be different from the username
Are you sure you want to use the password entered yes/no? yes
Re-enter Password :
What groups do you want this user to belong to? (Please enter a comma separated list, or leave blank for none)[ ]:
About to add user 'admin' for realm 'ManagementRealm'
Is this correct yes/no? yes
Added user 'admin' to file 'D:\projects\wildfly-10.0.0.Final\standalone\configuration\mgmt-users.properties'
Added user 'admin' to file 'D:\projects\wildfly-10.0.0.Final\domain\configuration\mgmt-users.properties'
Added user 'admin' with groups to file 'D:\projects\wildfly-10.0.0.Final\standalone\configuration\mgmt-groups.properties'
Added user 'admin' with groups to file 'D:\projects\wildfly-10.0.0.Final\domain\configuration\mgmt-groups.properties'
Is this new user going to be used for one AS process to connect to another AS process?
e.g. for a slave host controller connecting to the master or for a Remoting connection for server to server EJB calls.
yes/no? no
Для продолжения нажмите любую клавишу . . .

```

После добавления переходим по адресу <http://127.0.0.1:9991/>, где мы должны увидеть панель администратора:

The screenshot shows the WildFly administrator web console. The header includes the WildFly logo and a user menu with 'Messages: 0' and 'admin'. The navigation bar contains links for Home, Deployments, Configuration, Runtime, Access Control, and Patching. The main content area is organized into a grid of six sections, each with an icon, title, description, and a 'Start' button:

- Deployments**: Add and manage deployments. Includes a 'Deploy an Application' button and instructions: '1. Use the 'Add Deployment' wizard to deploy the application', '2. Enable the deployment'.
- Configuration**: Configure subsystem settings. Includes a 'Create a Datasource' button and instructions: '1. Select the Datasources subsystem', '2. Add a Non-XA or XA datasource', '3. Use the 'Create Datasource' wizard to configure the datasource settings'.
- Runtime**: Monitor server status. Includes a 'Monitor the Server' button and instructions: '1. Select the server', '2. View log files or JVM usage'.
- Access Control**: Manage user and group permissions for management operations. Includes an 'Assign User Roles' button and instructions: '1. Add a new user or group', '2. Assign one or more roles to that user or group'.
- Patching**: Manage WildFly patches.
- Need Help?**: General Resources and Get Help links.

## 4 Среда разработки

Будем рассматривать процесс разработки в среде IntelliJ Idea, которую мы можем скачать с [официального сайта](#).

## 5 База данных

База данных создается с помощью миграции. Файл миграции лежит в /migrations/.

Для работы с миграциями необходимо скачать Flyway. У этого приложения есть инструмент для работы из командной строки. Скачать его можно на [официальном сайте](#).

### Download

These are the downloads for the latest version of Flyway:

Client	Downloads	Source
Command-line Tool	flyway-commandline-4.0.1-windows-x64.zip flyway-commandline-4.0.1-linux-x64.tar.gz flyway-commandline-4.0.1-macosx-x64.tar.gz flyway-commandline-4.0.1.zip (without JRE) flyway-commandline-4.0.1.tar.gz (without JRE)	flyway-commandline-4.0.1-sources.jar
API	flyway-core-4.0.1.jar	flyway-core-4.0.1-sources.jar
Maven Plugin	flyway-maven-plugin-4.0.1.jar	flyway-maven-plugin-4.0.1-sources.jar
Gradle Plugin	flyway-gradle-plugin-4.0.1.jar	flyway-gradle-plugin-4.0.1-sources.jar
Ant Tasks	flyway-ant-4.0.1.zip flyway-ant-4.0.1.tar.gz	flyway-ant-4.0.1-sources.jar
SBT Plugin	flyway-sbt-4.0.1.jar	flyway-sbt-4.0.1-sources.jar

Older versions are available from [Maven Central](#)

Рассмотрим на примере для Windows. Скачиваем и распаковываем архив. Нам нужны будут 2 папки:

conf	06.05.2016 11:22	Папка с файлами	
drivers	06.05.2016 11:22	Папка с файлами	
jars	06.05.2016 11:22	Папка с файлами	
jre	06.05.2016 11:22	Папка с файлами	
lib	06.05.2016 11:22	Папка с файлами	
sql	29.05.2016 21:13	Папка с файлами	
flyway	06.05.2016 11:21	Файл	2 КБ
flyway.cmd	06.05.2016 11:21	Сценарий Windo...	2 КБ
LICENSE.txt	06.05.2016 11:21	Файл "TXT"	1 КБ
LICENSES-THIRD-PARTY.txt	06.05.2016 11:21	Файл "TXT"	106 КБ
README.txt	06.05.2016 11:21	Файл "TXT"	1 КБ

Теперь нужно настроить адрес до БД. Заходим в распакованную папку и открываем файл `conf/flyway.conf`.

Находим строчку «`flyway.url=`». Выше, в комментариях есть примеры того, что надо писать для разных БД. Например, `mariadb`:

```
flyway.url=jdbc:mariadb://127.0.0.1:3306/test
```

Теперь копируем в папку `sql/` файл миграции из проекта `/migrations/V1__Repository_migration.sql`.

Запускаем `cmd`, переходим в распакованную папку, например, `flyway-4.0.1` и выполняем команду:

```
flyway migrate
```

У нас попросят вести пользователя для БД и его пароль, после чего будет применена миграция:

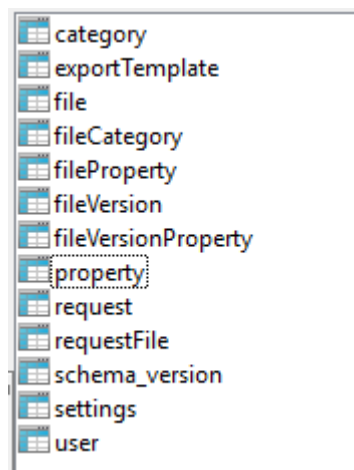
```
C:\flyway-4.0.1>flyway migrate
Flyway 4.0.1 by Boxfuse

Database user: root
Database password:
Database: jdbc:mysql://127.0.0.1:3306/test (MySQL 10.0)
Successfully validated 1 migration (execution time 00:00.006s)
Creating Metadata table: `test`.`schema_version`
Current version of schema `test`: << Empty Schema >>
Migrating schema `test` to version 1 - Repository migration
Successfully applied 1 migration to schema `test` (execution time 00:01.836s).

C:\flyway-4.0.1>_
```

Открываем БД и видим созданные таблицы:



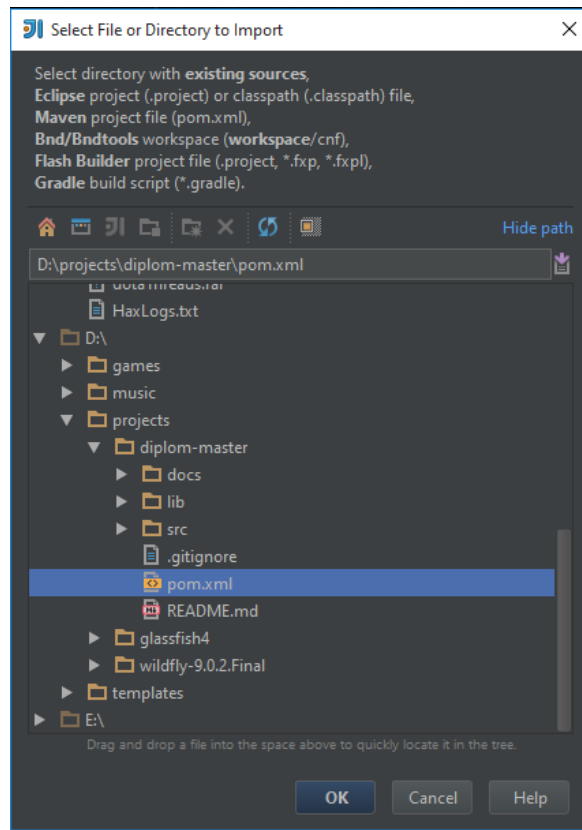


Логическая схема базы данных приведена на рисунке:



## 6 Создание проекта в IntelliJ Idea

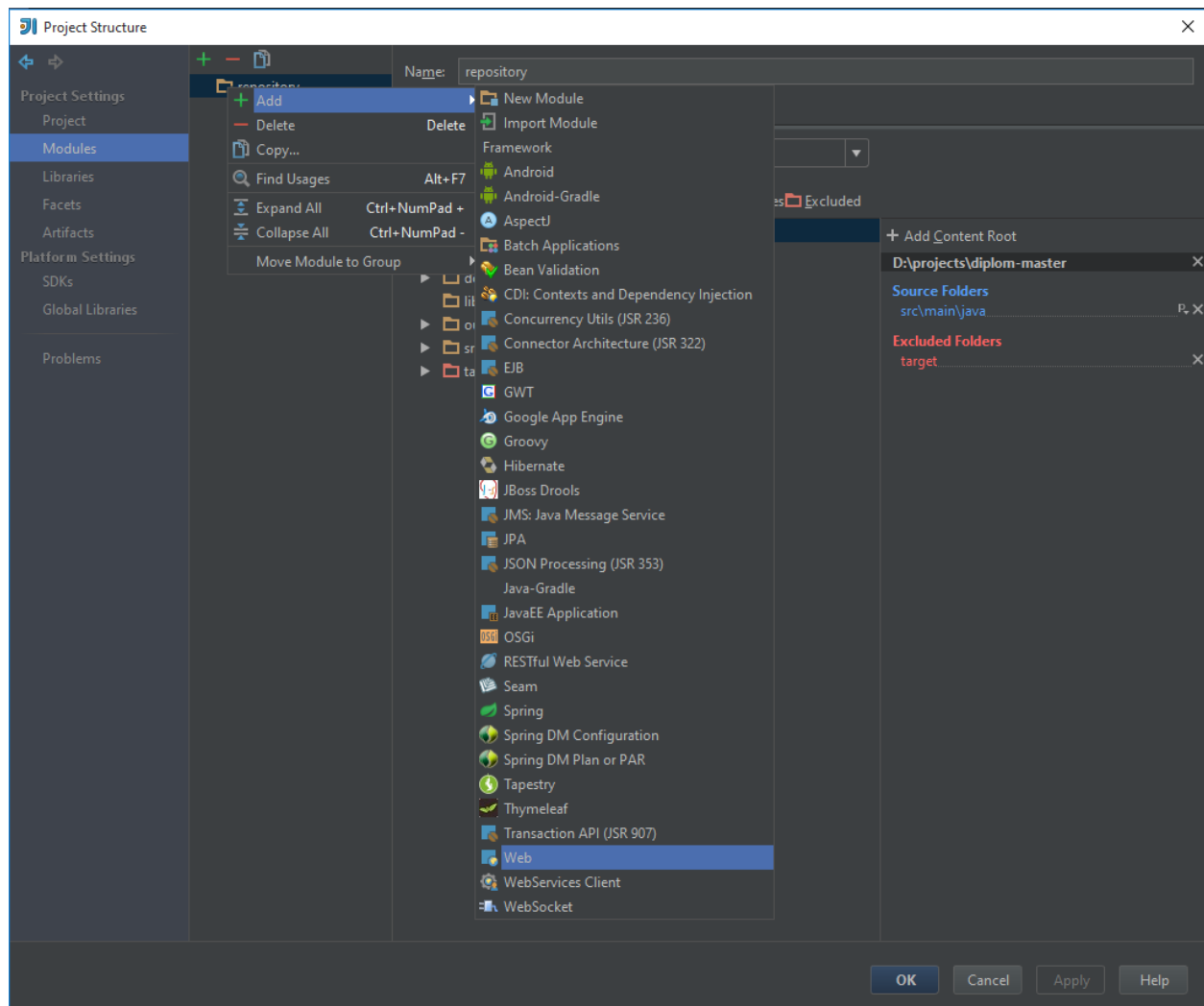
Для создания проекта нажимаем «Import project» и выбираем путь до исходных файлов, до файла pom.xml.



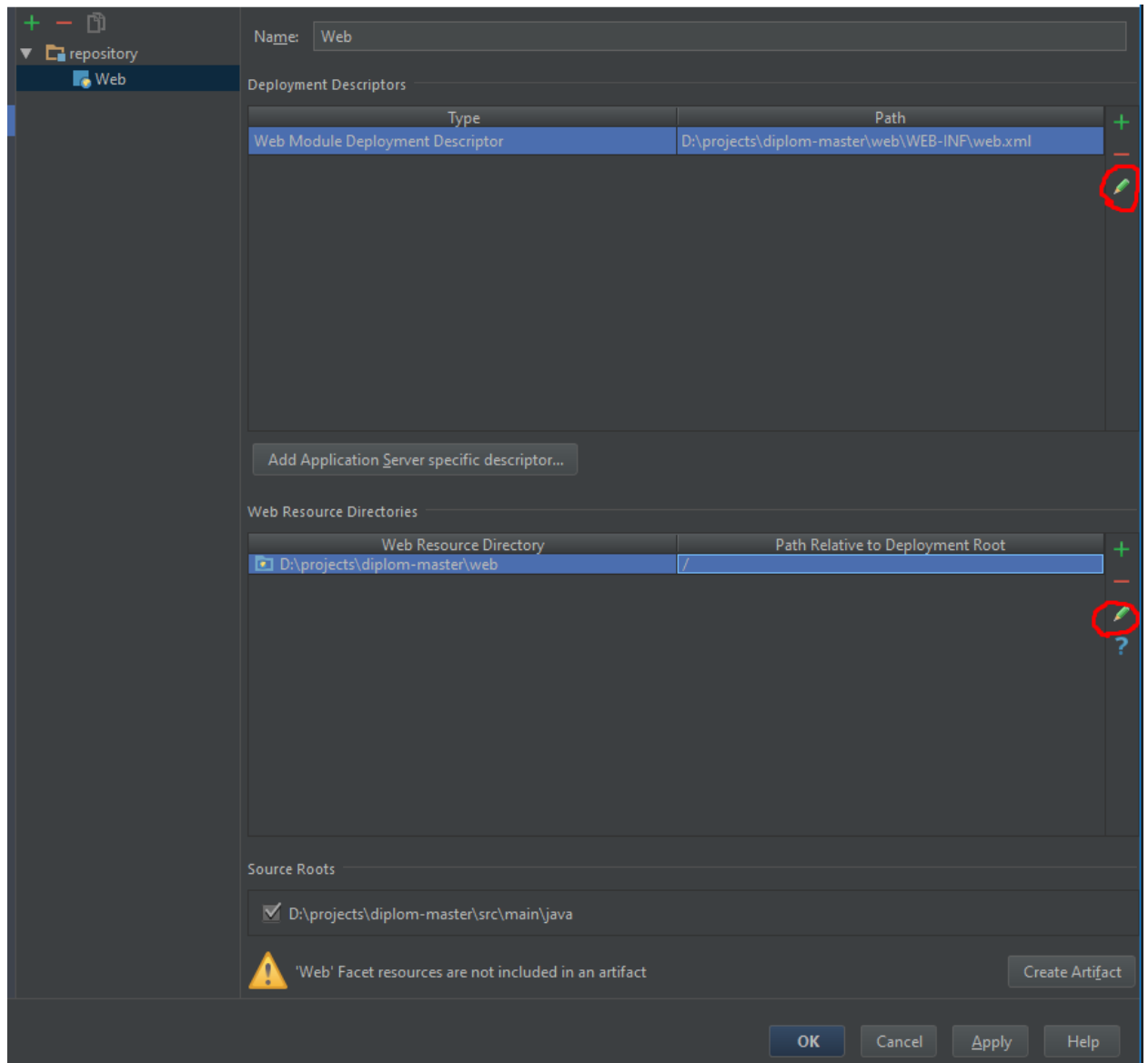
Жмем «ok», далее оставляем все без изменений, нажимаем «next», «next», «finish».

### 6.1 Создание модулей

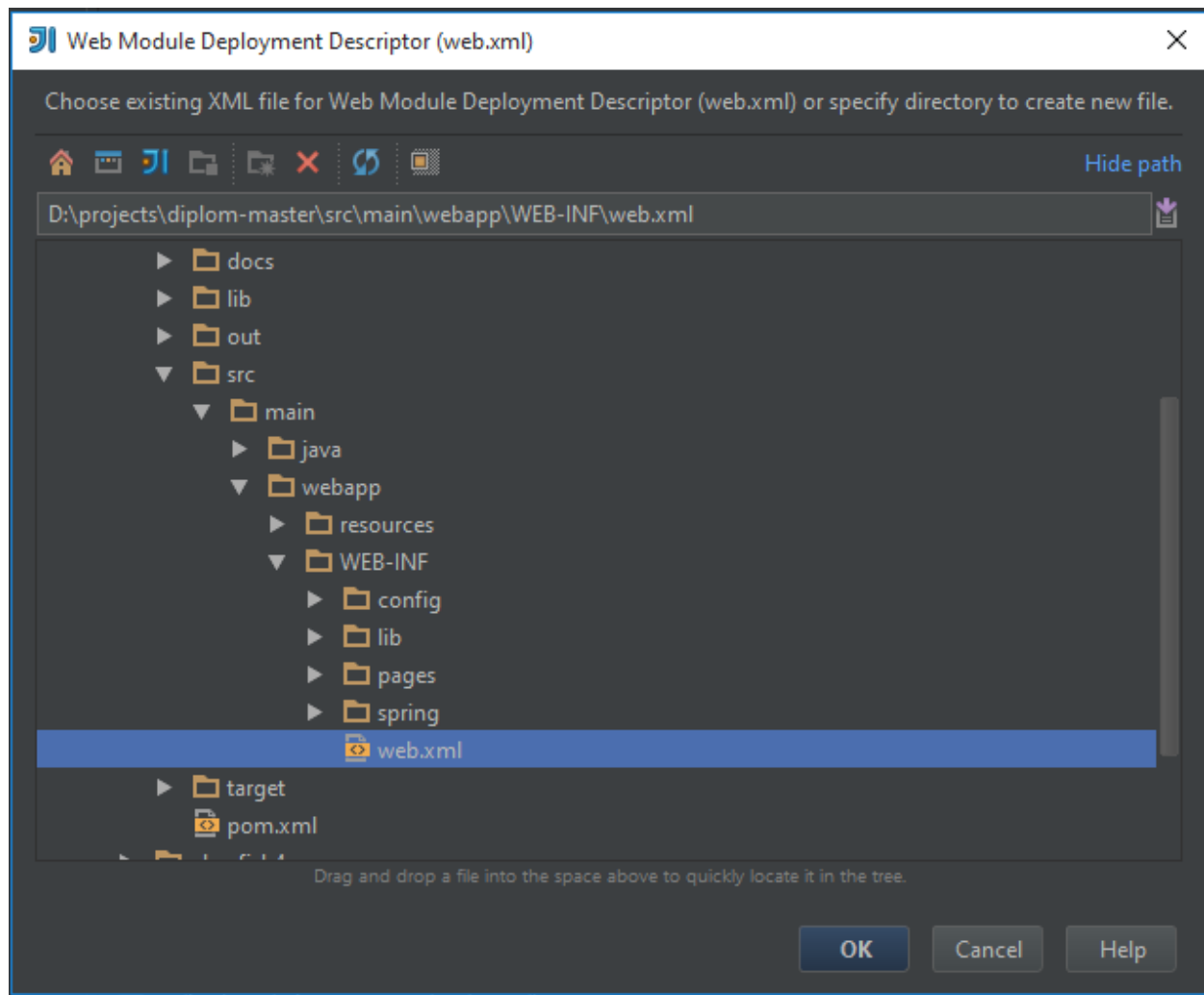
Нажимаем «File» -> «Project Structure» -> «Modules», правой кнопкой по «repository» -> «add» -> «Web».



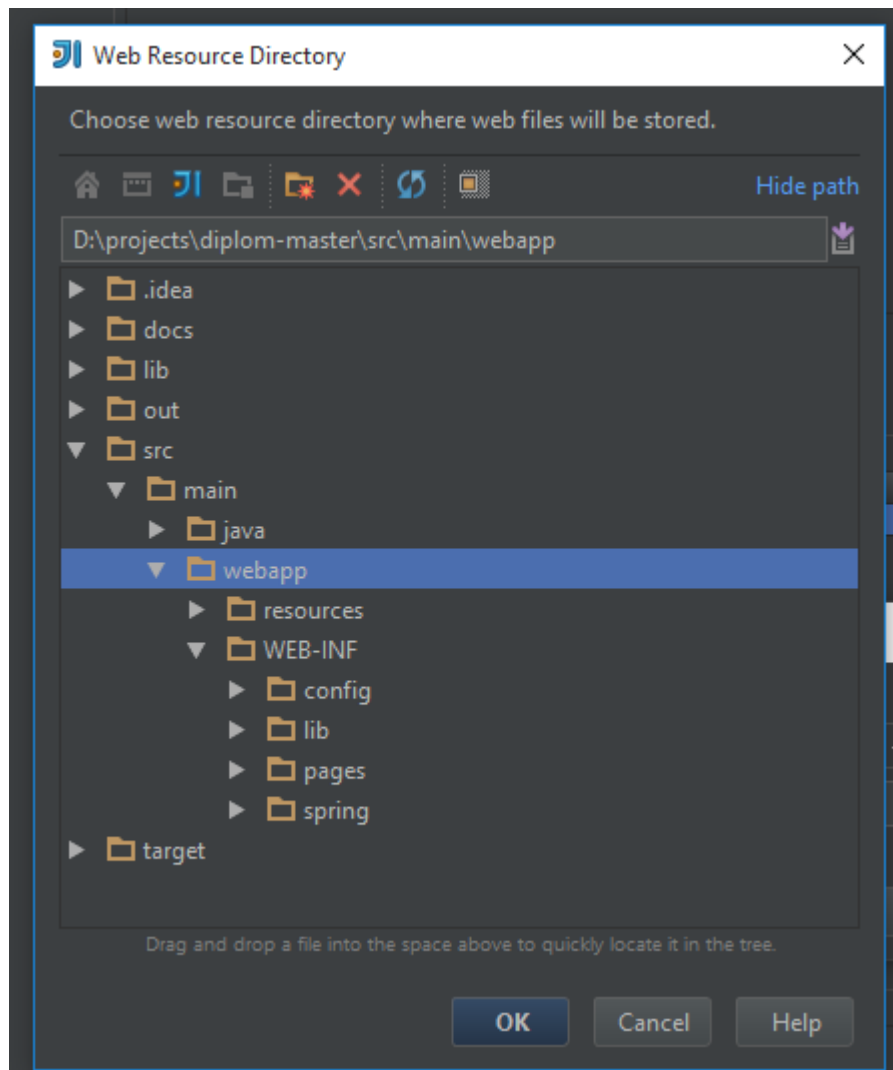
Далее необходимо настроить пути до папки webapp.



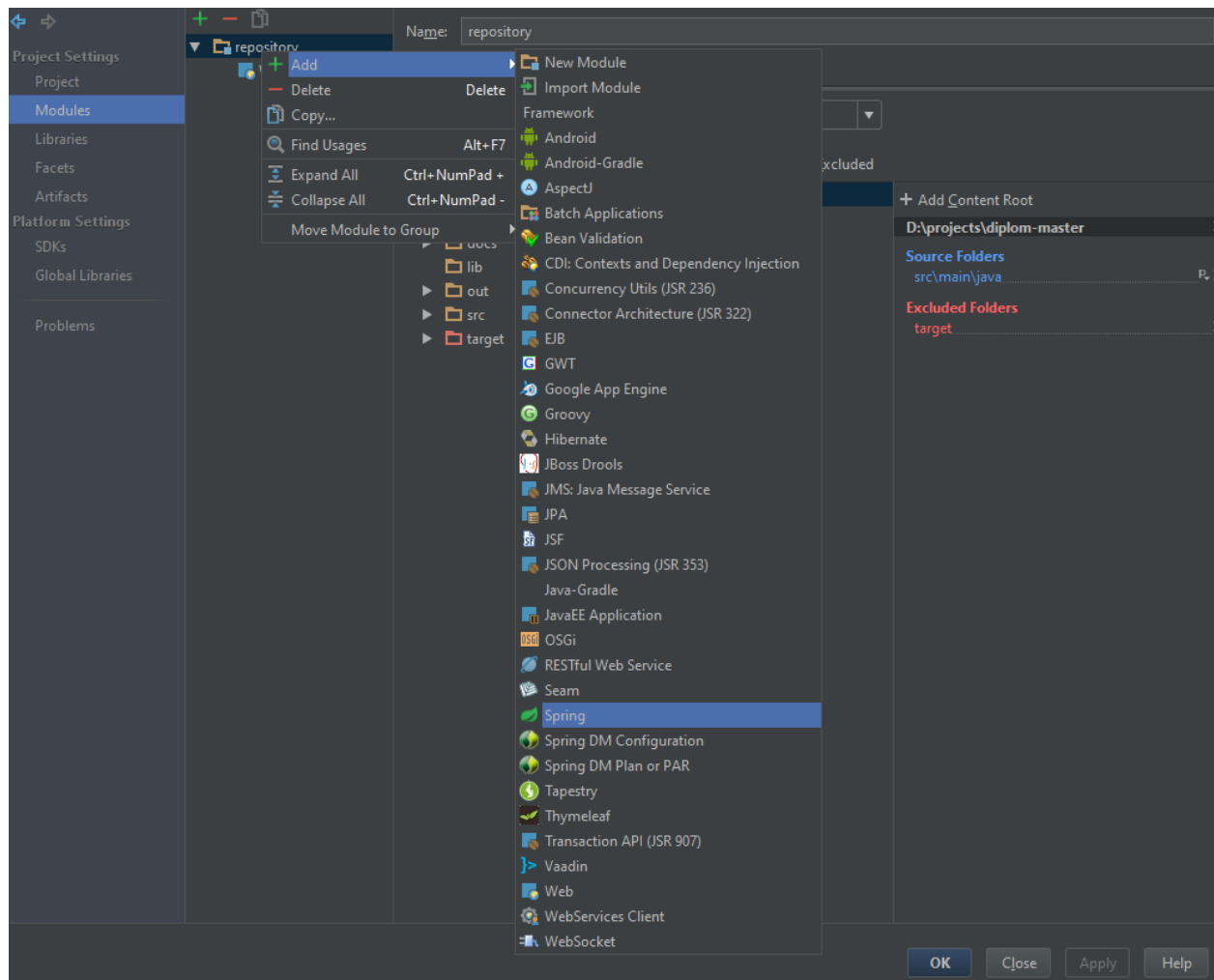
В верхнем пути выбираем путь до web.xml:



В нижнем – путь до папки webapp:

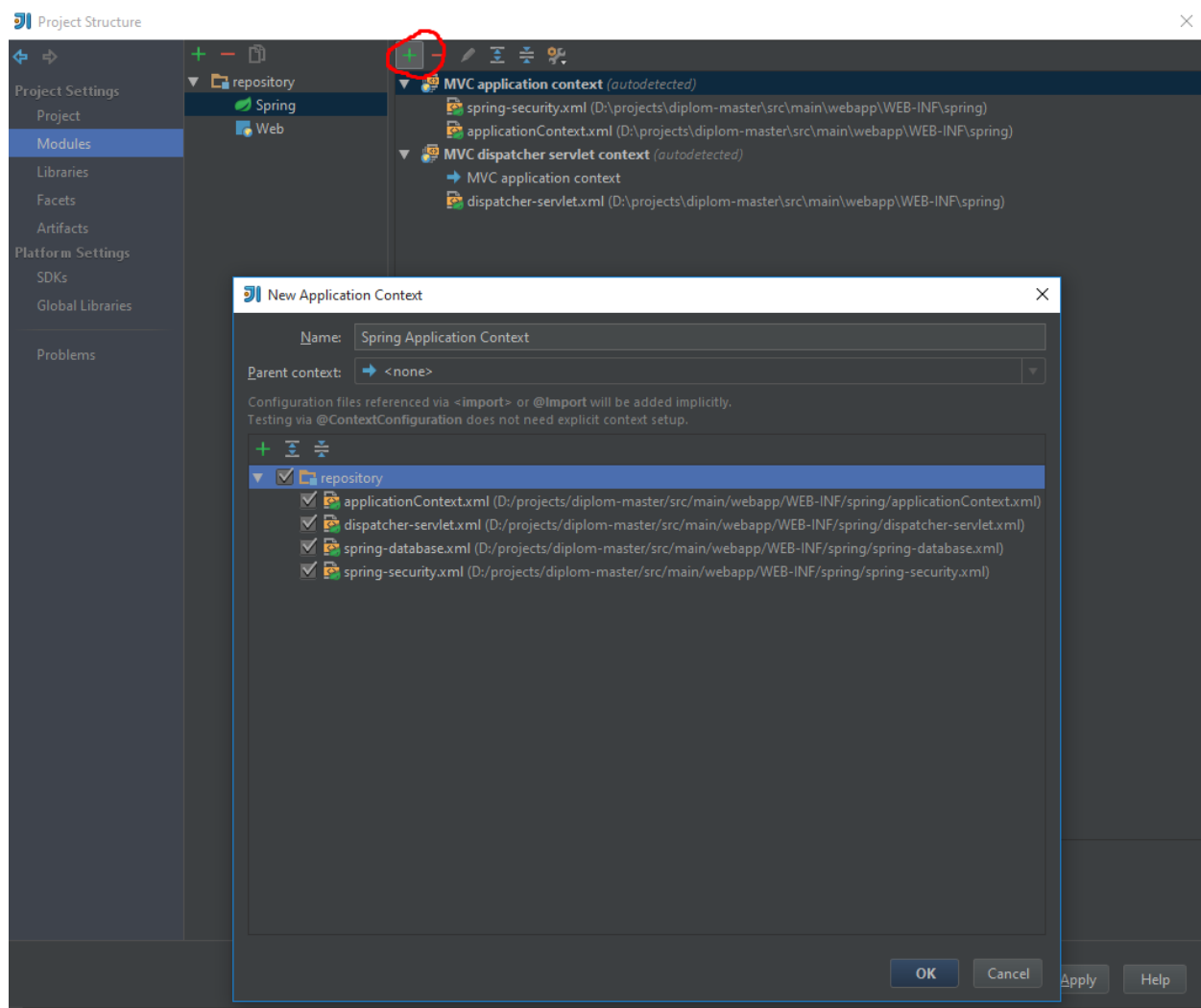


Добавляем еще 1 модуль Spring



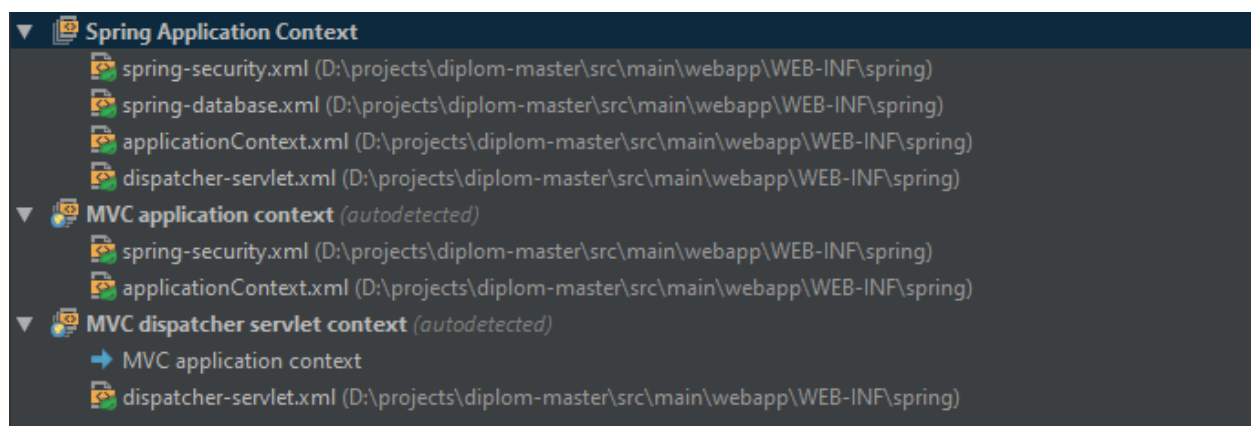
Нажимаем на «+»





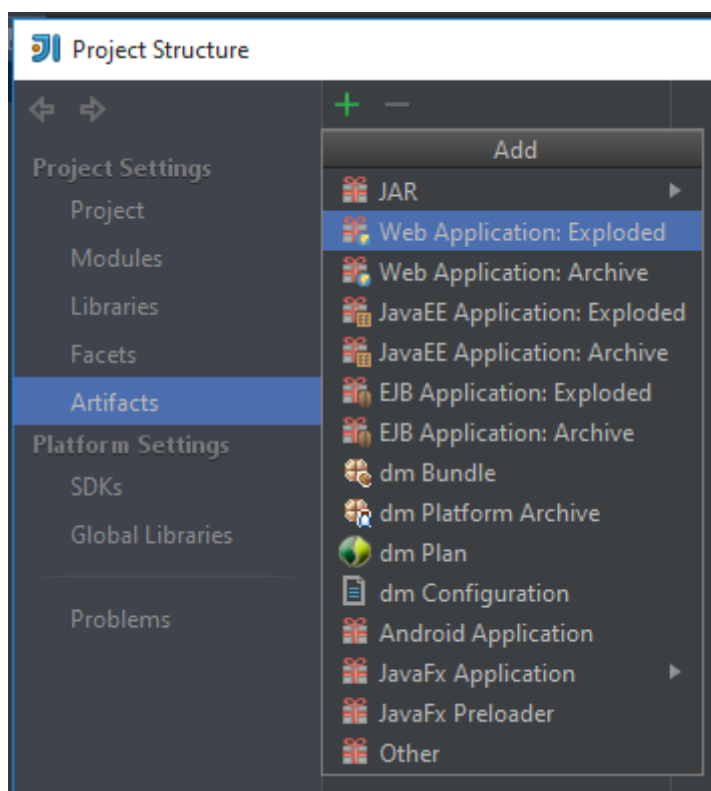
В появившемся окне выбираем все файлы и нажимаем «ок». Далее еще раз «ок».

В результате у модуля Spring должна быть следующая конфигурация

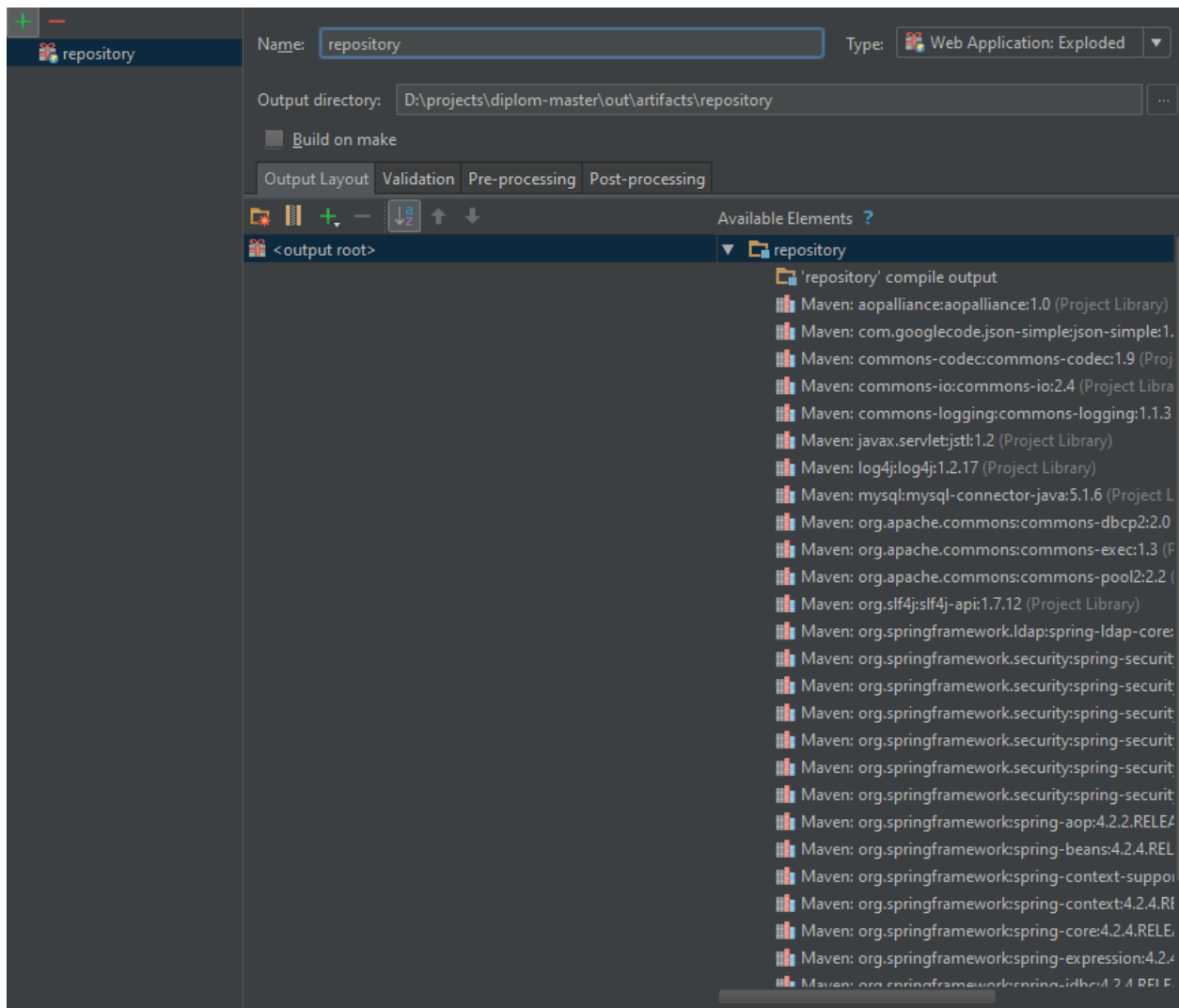


## 6.2 Создание артефакта

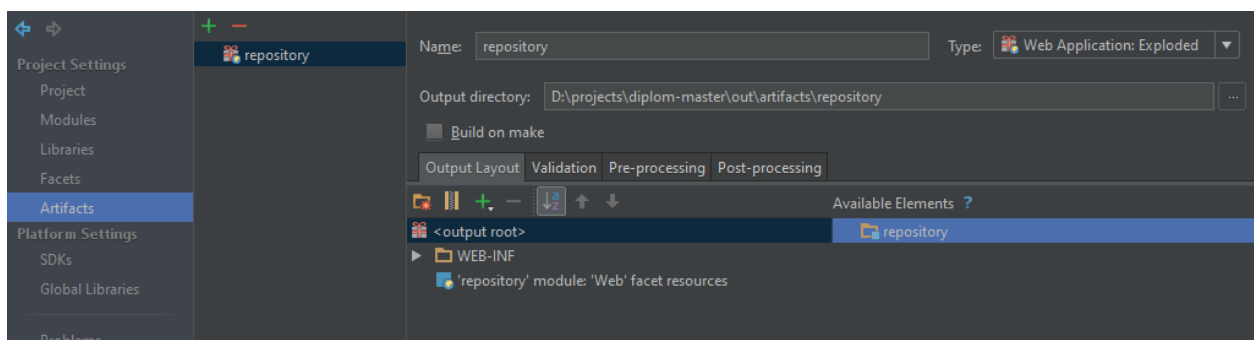
Необходимо создать артефакт, который будет деплоиться на сервер приложений. Для этого, нажимаем «File» -> «Project Structure» -> «Artifacts» -> «+» -> «Web Application: Exploded».



Вводим любое название в поле Name



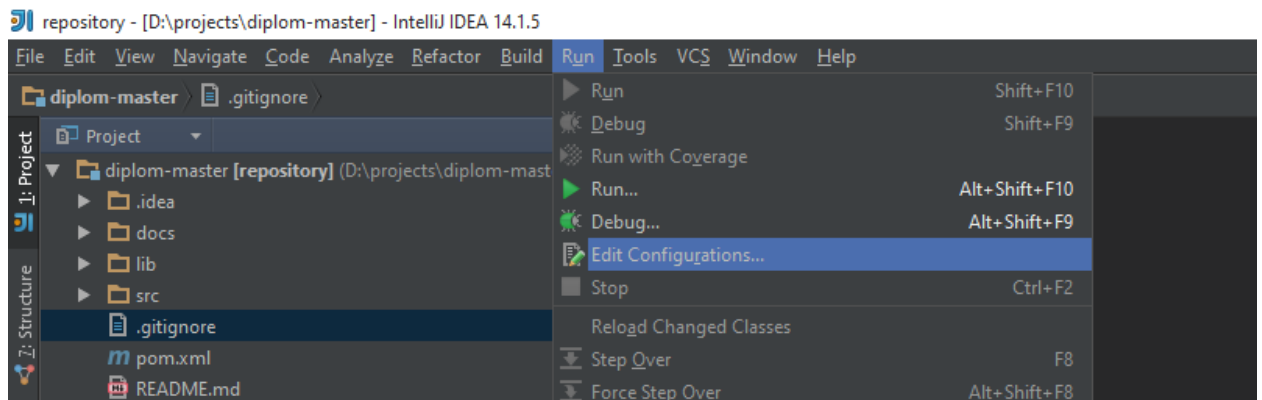
Нажимаем на «repository» справа правой кнопкой и, затем, «Put into output root», после чего библиотеки справа должны переместиться в артефакт:



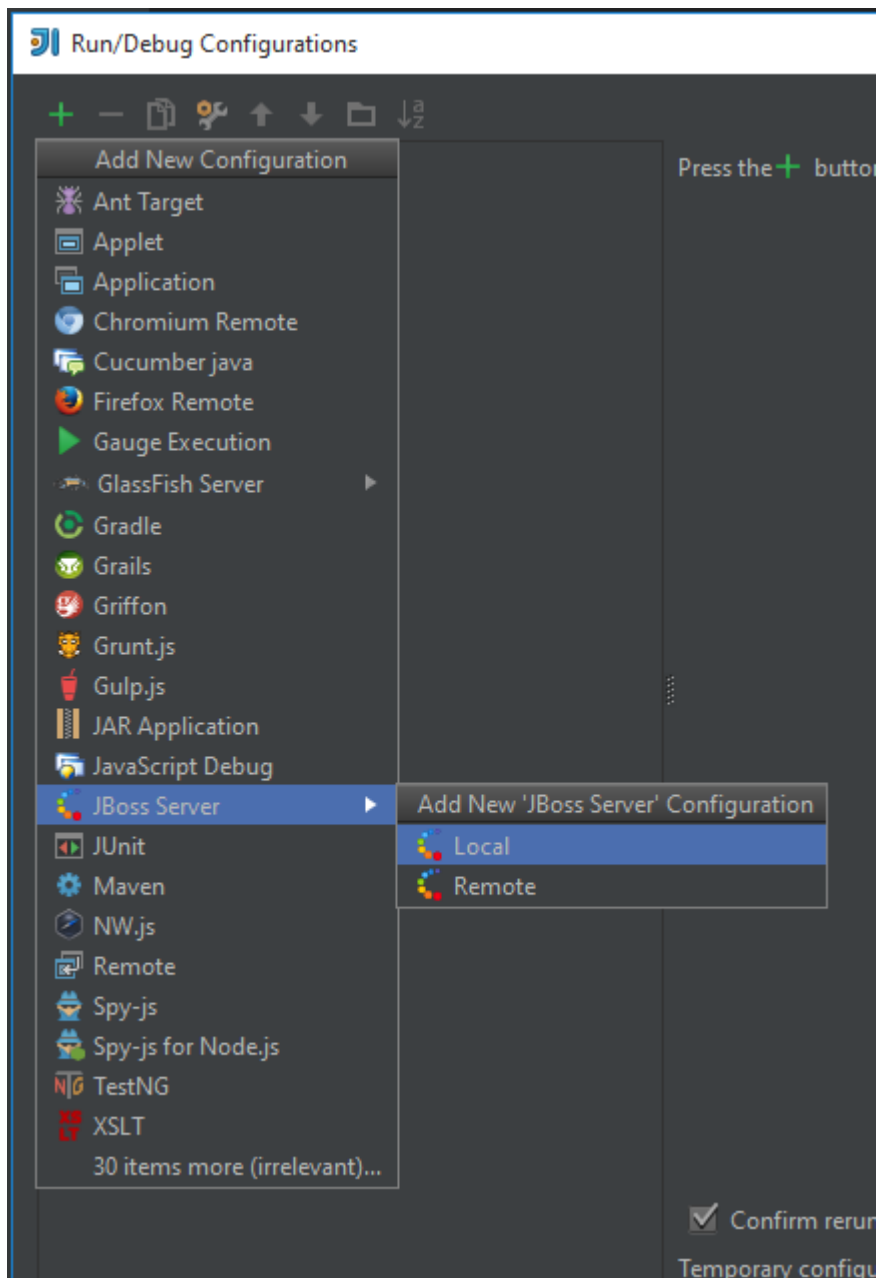
Артефакт создан, нажимаем «ок».

## 6.3 Создание конфигурации запуска

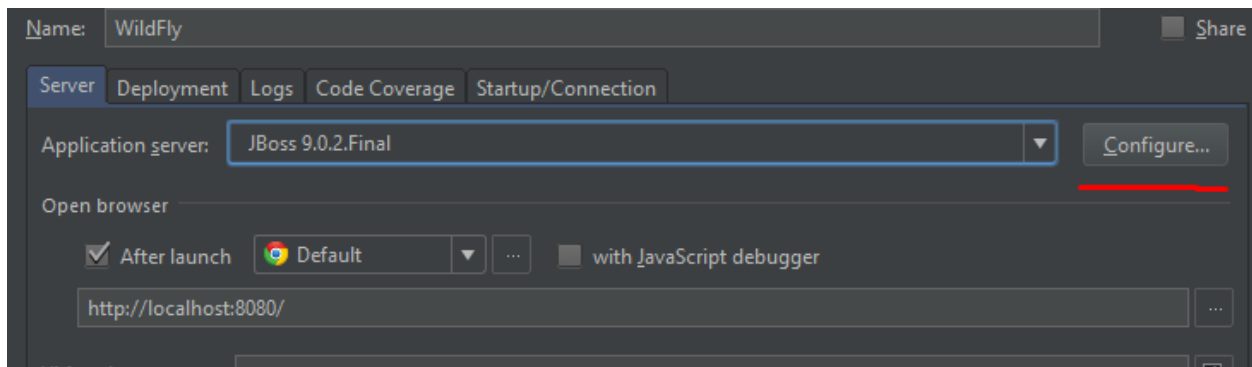
Далее, для запуска приложения, необходимо создать конфигурацию для запуска и добавить в Idea скаченный ранее сервер приложений WildFly. Для этого, нажимаем «Run» -> «Edit configuration».



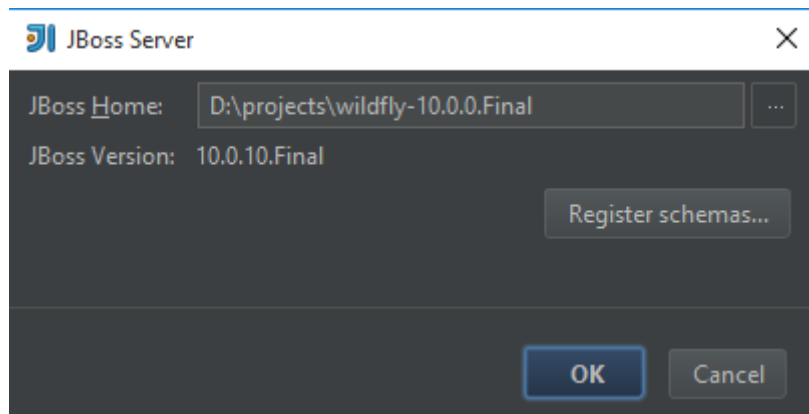
Далее, «+» -> «JBoss Server» -> «local»



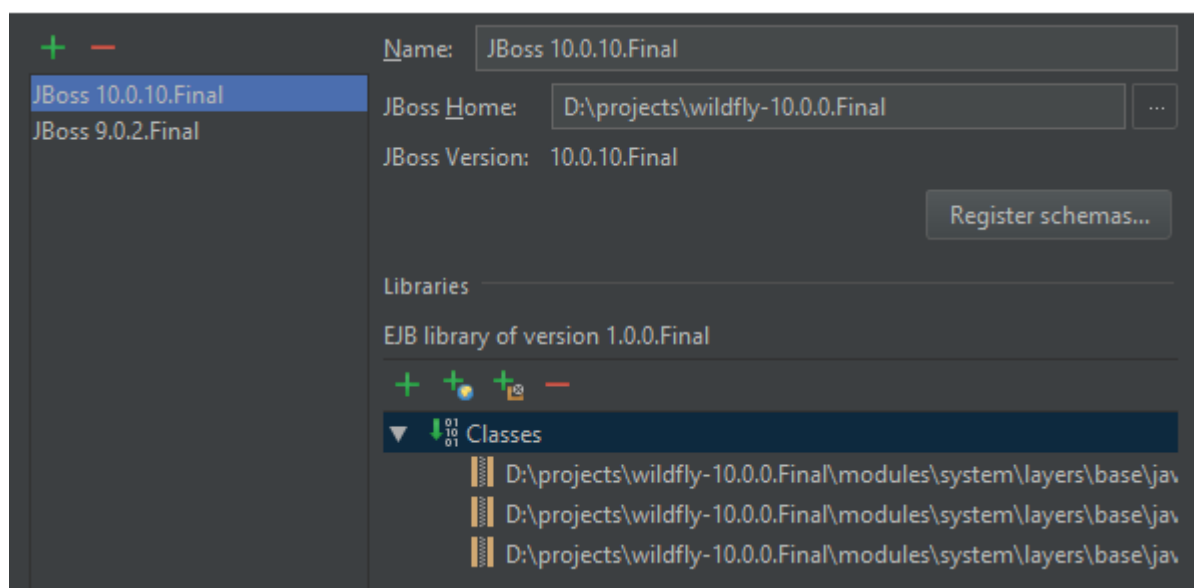
Нажимаем «Configure»



В появившемся окне нажимаем «+», выбираем путь до папки с сервером, в который распаковывали его в пункте 3 и нажимаем «ok».

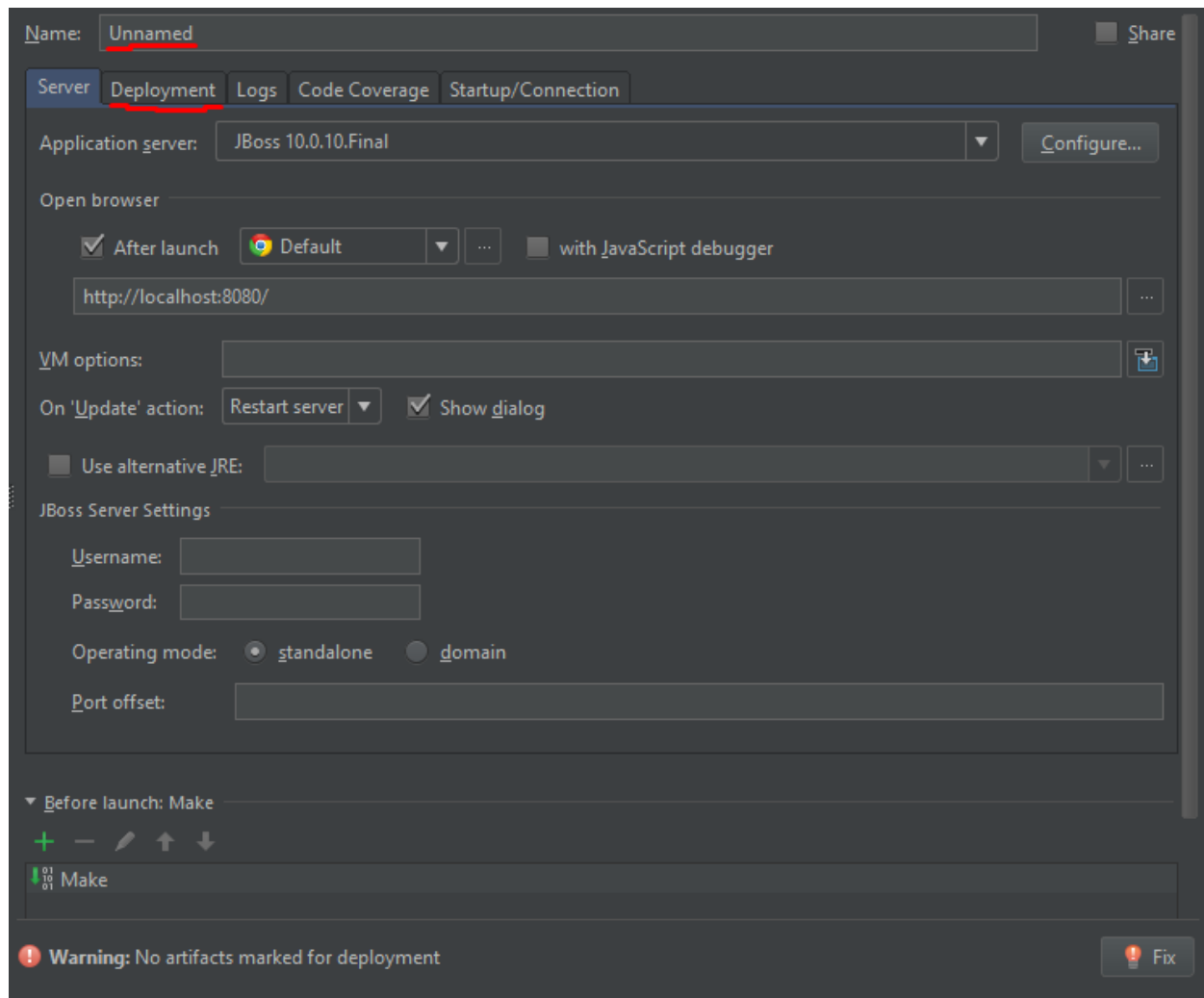


В списке серверов должен появиться этот сервер. Выбираем его и нажимаем «ok».

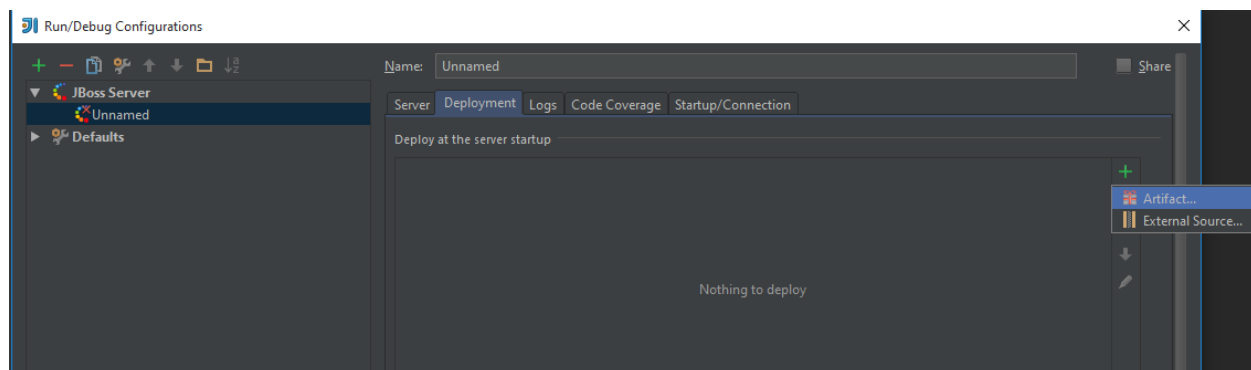


Далее продолжаем заполнять настройки конфигурации для запуска

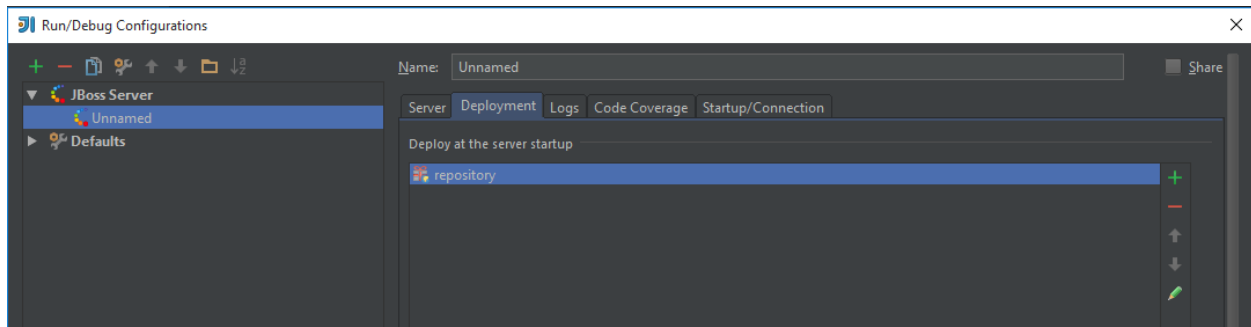
В поле Name можно ввести любое название. Далее переходим во вкладку «Deployment»:



Нажимаем на «+» справа -> «Artifact...»



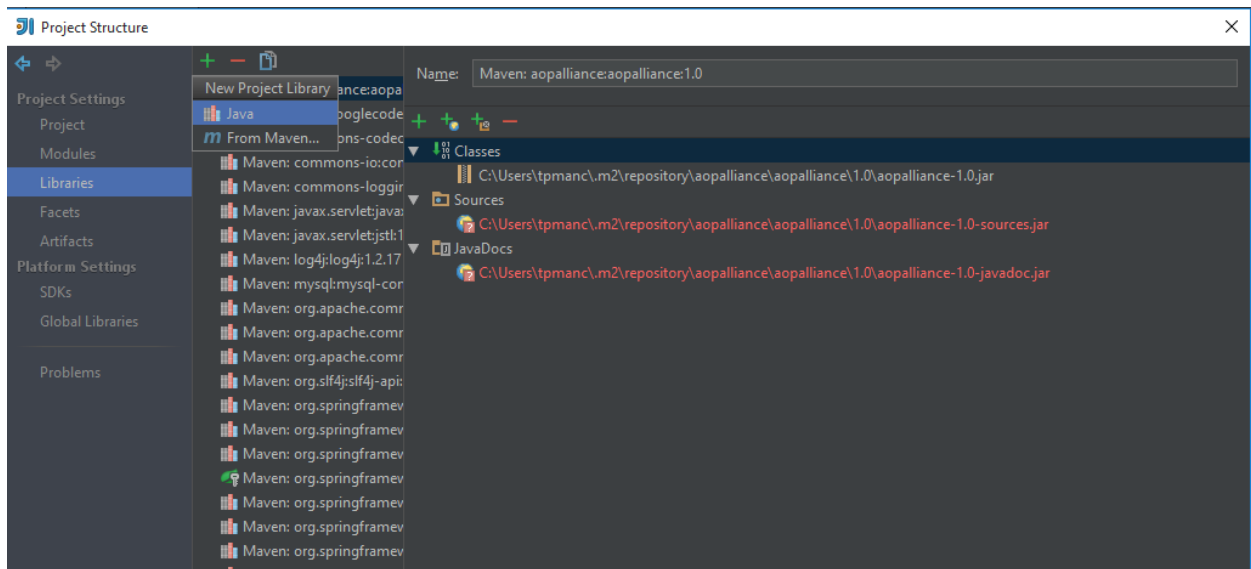
После этого должен появиться созданный нами в пункте 4 артефакт.



Нажимаем «ок», после чего конфигурация будет создана.

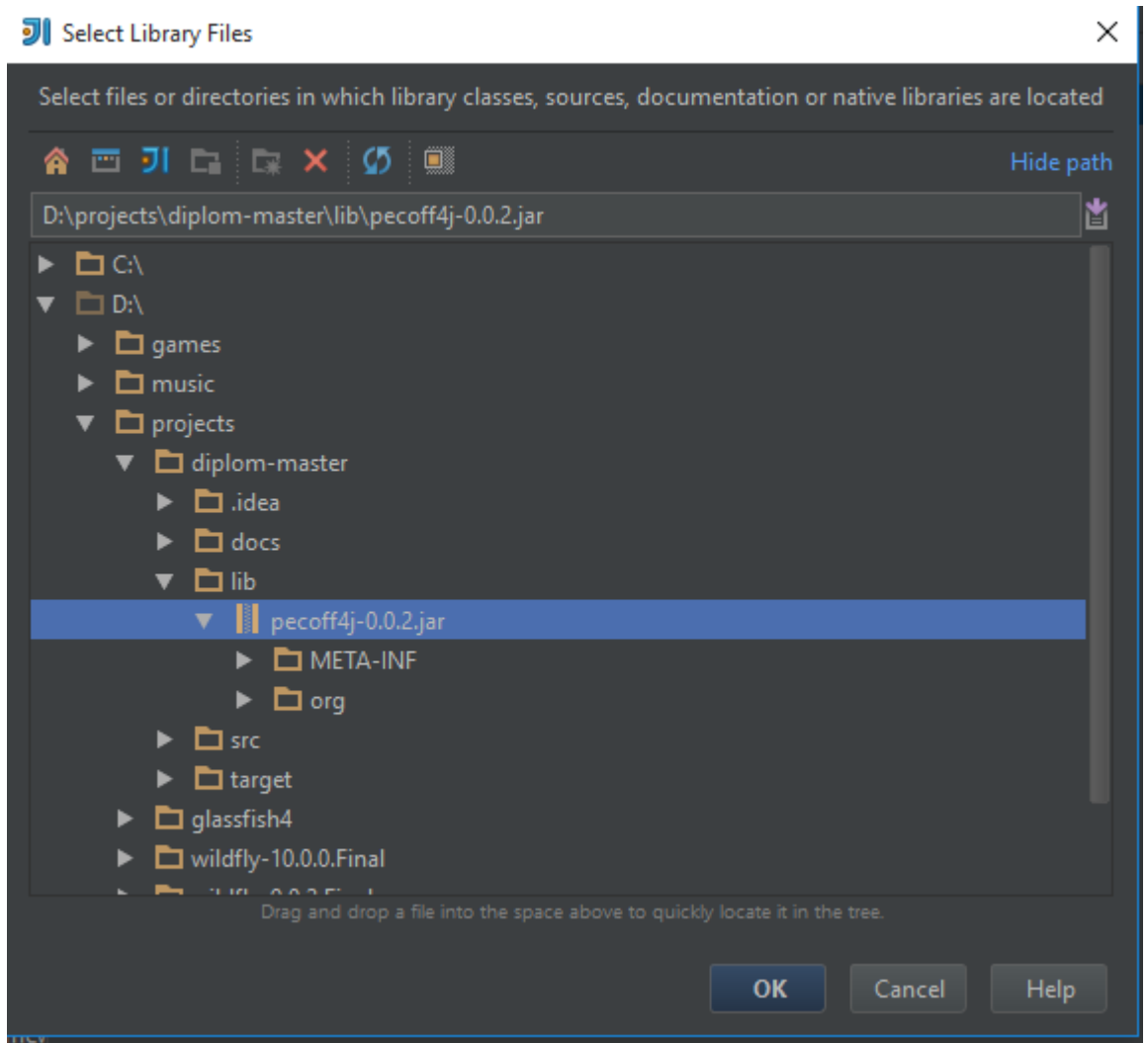
## 6.4 Добавление библиотеки

Необходимо добавить 1 библиотеку вручную. Нажимаем «File» -> «Project Structure» -> «Libraries» -> «+» -> «Java».



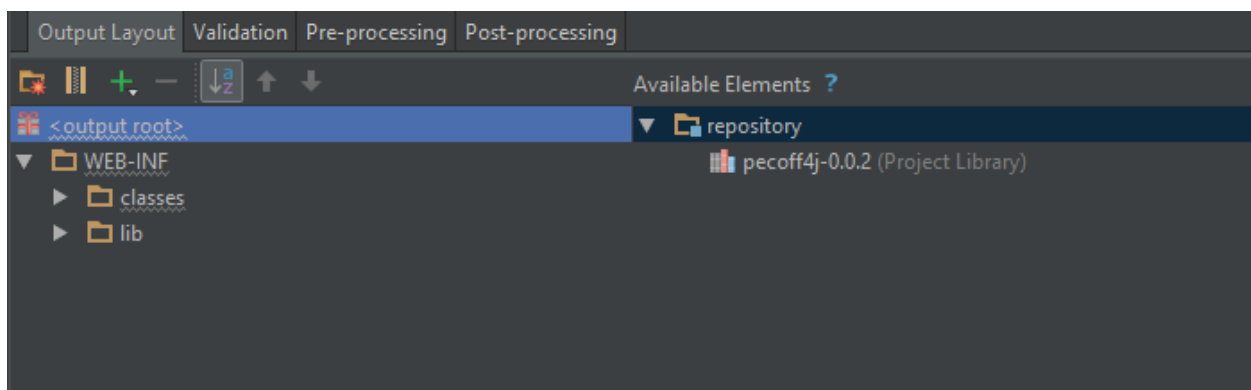
В появившемся окне выбираем путь до папки с проектом, файл lib/pesoff4j-0.0.2.jar и нажимаем «ок».

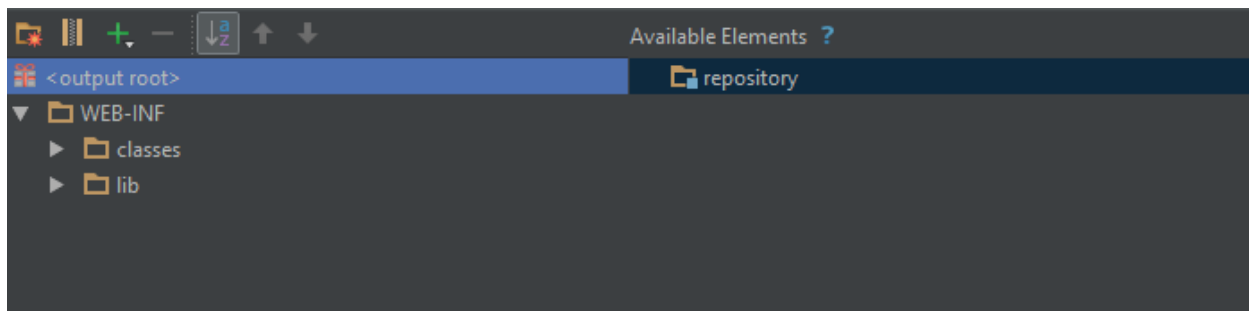




После добавления библиотеки ее необходимо добавить в артефакт. Также, любую другую библиотеку, которую мы добавили в проект, необходимо добавлять в артефакт.

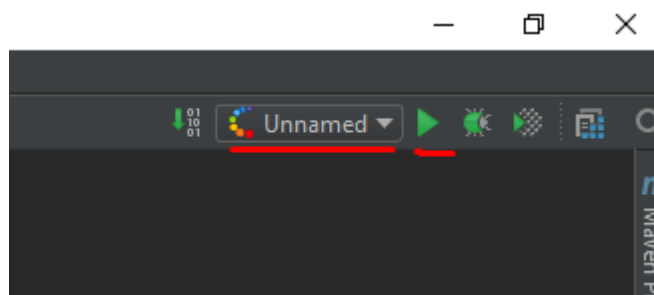
«File» -> «Project Structure» -> «Artifacts» и двойным кликом по библиотеке перемещаем ее влево.





## 6.5 Запуск

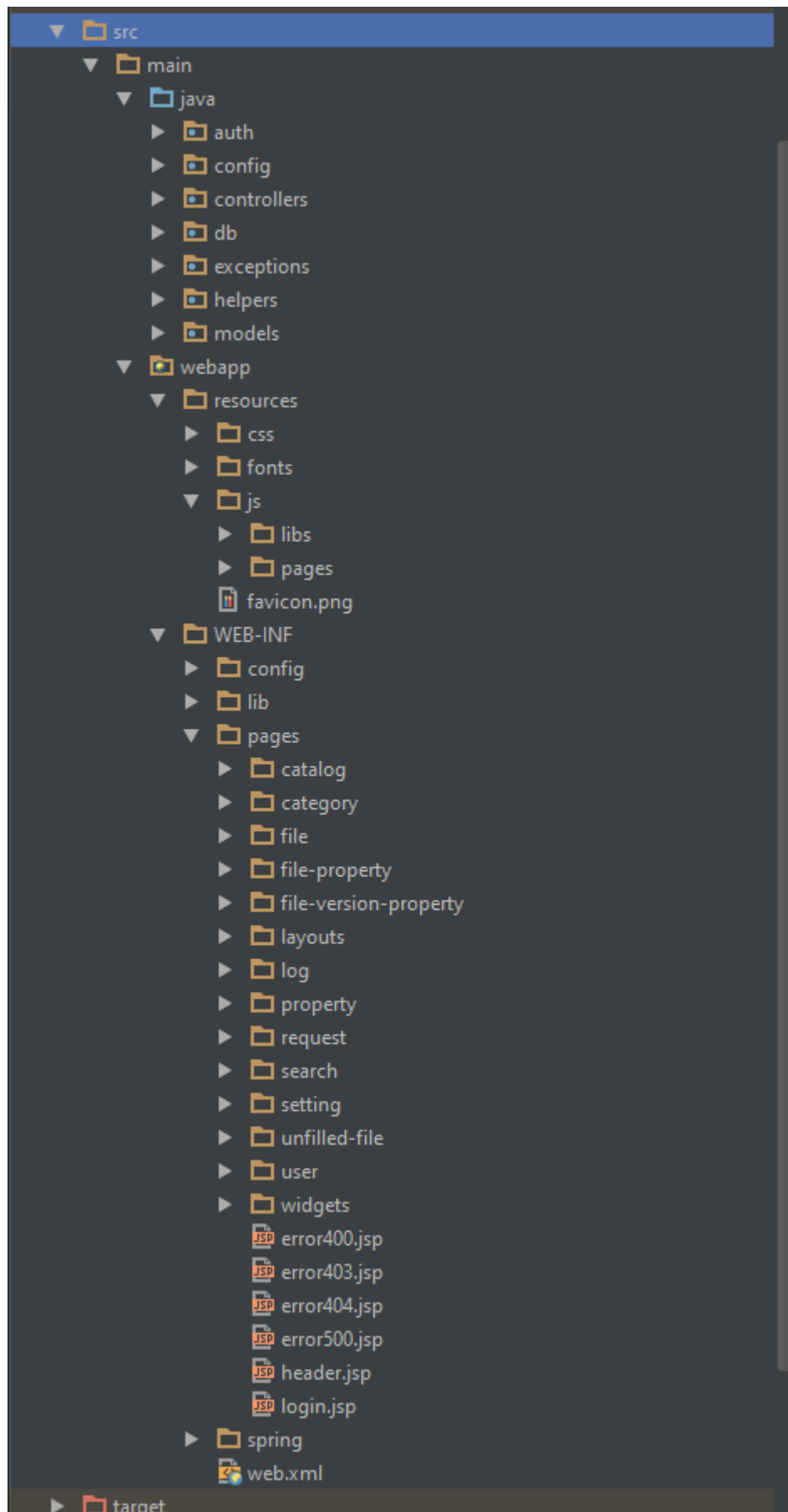
Теперь можно запустить приложение. Справа сверху выбираем название нужной конфигурации и жмем пуск.



Idea сама запустит сервер приложений а задеплоит артефакт на него. При успешном запуске в консоли выведется следующий текст:

```
public java.lang.String controllers.VersionPropertyController.fileAddPropertyHandler(int,int)
12:45:44,811 INFO [org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping] namespace 'dispatcher-servlet': startup date [Sun May 29 12:45:44 MSK 2016]; parent: Root WebApplicationContext
12:45:44,839 INFO [org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerMapping] namespace 'dispatcher-servlet': startup date [Sun May 29 12:45:44 MSK 2016]; parent: Root WebApplicationContext
12:45:44,901 INFO [org.springframework.web.servlet.mvc.method.annotation.ExceptionHandlerExceptionResolver] exceptionHandlerController
12:45:44,958 INFO [org.springframework.web.servlet.DispatcherServlet] (ServerService Thread Pool -- 95) WFLYUT0: ServletContext initialized
12:45:44,959 INFO [org.wildfly.extension.undertow] (ServerService Thread Pool -- 95) WFLYUT0: ServletContext initialized
12:45:44,983 INFO [org.jboss.as.server] (management-handler-thread - 3) WFLYSRV0010: Deployed "ROOT.war"
[2016-05-29 12:45:45,074] Artifact repository: Artifact is deployed successfully
[2016-05-29 12:45:45,074] Artifact repository: Deploy took 3 865 milliseconds
12:45:45,325 INFO [controllers.LoginController] (default task-2) This is an info log entry
12:45:46,418 INFO [controllers.LoginController] (default task-5) This is an info log entry
```

## 7 Структура проекта






Весь код лежит в папке `src/main/java`.

- auth – классы для работы с Active Directory
- controllers - контроллеры
- models - модели
- helpers – вспомогательные классы

Виды лежат в src/webapp/WEB-INF/pages/название модели/.

При первом запуске в домашней папке пользователя создается папка repository, в которой создаются файлы конфигурации БД, Active Directory и логирования.

 active-directory.properties	14.05.2016 17:00	Файл "PROPERTIES"	1 КБ
 database.properties	29.05.2016 19:47	Файл "PROPERTIES"	1 КБ
 log4j.properties	14.05.2016 17:00	Файл "PROPERTIES"	1 КБ

## 7.1 Контроллеры

Контроллеры должны помечаться аннотацией @Controller.

Для того, чтобы привязать метод контроллера к какому-либо URL, нужно добавить аннотацию, например,

```
@RequestMapping(value = {"/catalog", "/"}, method =RequestMethod.GET)
```

Каждый метод контроллера должен возвращать строку, в которой будет путь до нужного файла jsp вида относительно папки src/main/webapp/WEB-INF/pages. Например, метод возвращает строку "catalog/catalog". Тогда будет использоваться файл src/main/webapp/WEB-INF/pages catalog/catalog.jsp.

Для того, чтобы передавать данные в вид, нужно в список входных параметров метода контроллера добавить Model model.

```
public String users(
    @RequestParam(value="categoryId", required=false, defaultValue = "0")
    int categoryId,
    @RequestParam(value="page", required=false, defaultValue = "1") int
    page,
    Principal principal,
    Model model
)
```

Теперь, для передачи данных в шаблон пишем

```
model.addAttribute("requestCount", requestCount);
```

Использовать это в шаблоне можно так:

```
<h2>${requestCount}</h2>
```

## 7.2 Модели

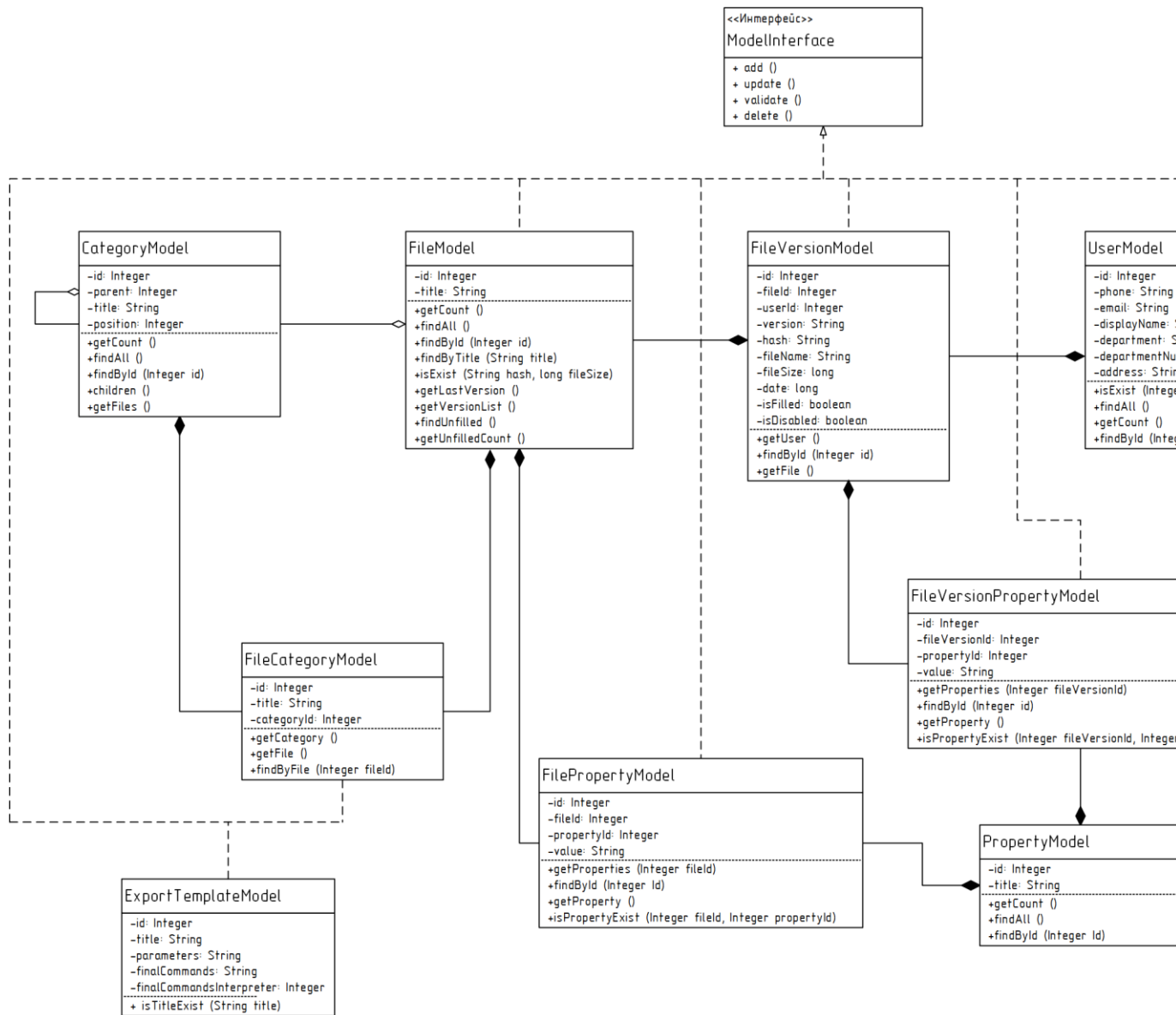
Все модели реализуют интерфейс `ModelInterface`. Необходимо реализовать методы: `add()`, `validate()`, `delete()`, `update()`.

Для работы в БД пишем:

```
NamedParameterJdbcTemplate template = new
NamedParameterJdbcTemplate(Database2.getInstance().getBds());
MapSqlParameterSource parameters = new MapSqlParameterSource();
parameters.addValue("id", id);
List<Map<String, Object>> rows = template.queryForList(getById, parameters);
for (Map row : rows) {
    Integer fileId = (Integer) row.get("id");
    String title = (String) row.get("title");
    return new FileModel(fileId, title);
}
```

Здесь создается шаблон и набор параметров, которые мы хотим подставить в SQL запрос.

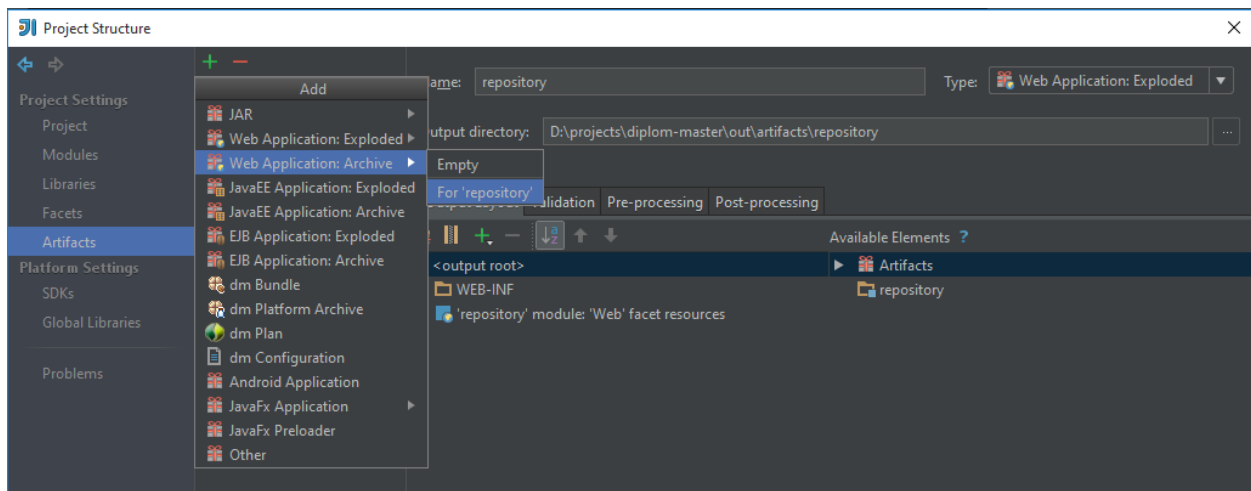
Диаграмма классов:



## 8 Билд

В результате билда нам нужно получить war архив. Для этого, надо создать еще один артефакт:

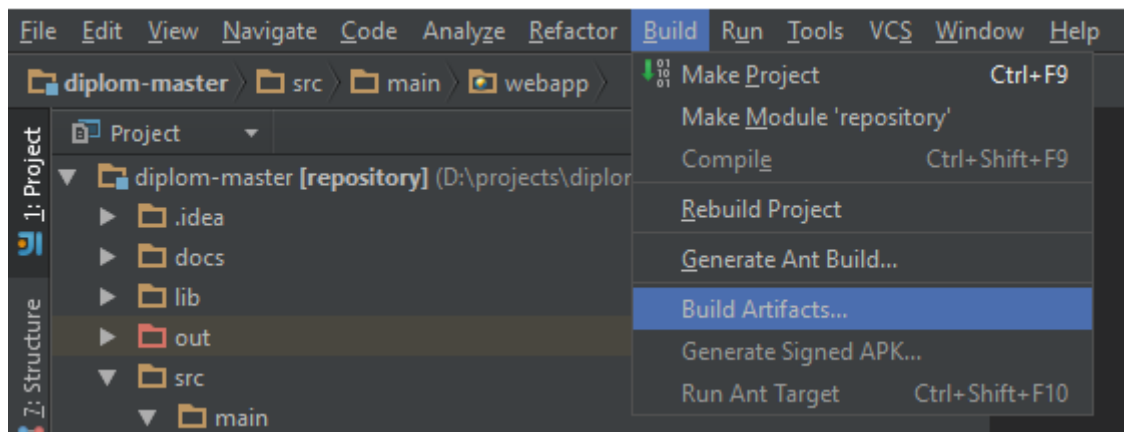
«File» -> «Project Structure» -> «Artifacts» -> «+» -> «Web Application: Archive»



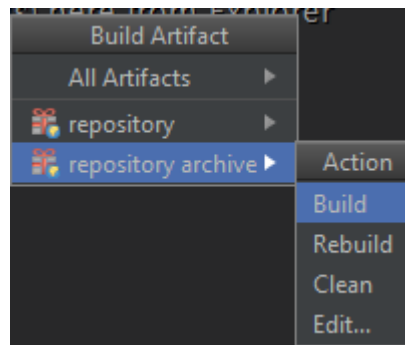
Обязательно выбираем For «repository». Вместо repository может быть другое название, если предыдущий артефакт был назван по-другому. Нажимаем «For «repository»», создается артефакт и нажимаем «ок», чтобы сохранить его.

Теперь сам билд:

Кнопка сверху «Build» -> «Build Artifacts»



В появившемся окошке выбираем только что созданный артефакт и Build.



Результат билда написан в самом низу:

```
Compilation completed successfully in 1s 852ms (moments ago)
```

Идем в папку с проектом, out/artifacts/repository\_archive и видим нужный нам war файл:

тер > Локальный диск (D:) > projects > diplom-master > out > artifacts > repository_archive			
Имя	Дата изменения	Тип	Размер
repository archive.war	29.05.2016 20:27	Файл "WAR"	25 959 КБ

Его можно деплоить на сервер приложений.

## 9 Работа с WildFly

### 9.1 Деплой на сервер приложений

Должен быть запущен сервер приложений. Пример запуска написан в пункте 3. Если коротко, то нужно запустить bin/standalone.bat.

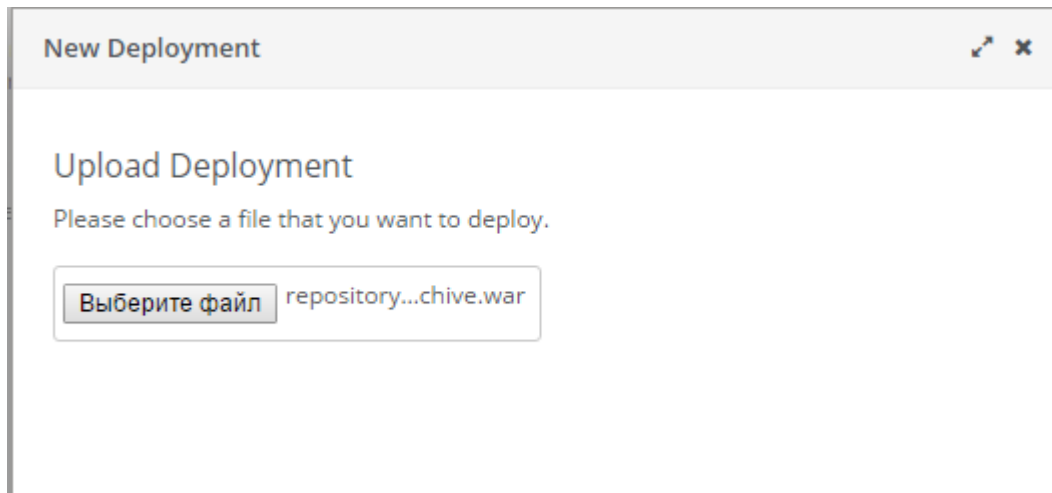
Заходим в панель администратора. Ее адрес пишется в консоль, в самом конце, при запуске сервера.

```
20:34:28,286 INFO [org.jboss.ws.common.management] (MSC service thread 1-2) JBWS022052: Starting JBossWS 5.1.3.Final (Apache CXF 3.1.4)
20:34:28,523 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0060: Http management interface listening on http://127.0.0.1:9991/management
20:34:28,525 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0051: Admin console listening on http://127.0.0.1:9991
20:34:28,526 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: WildFly Full 10.0.0.Final (WildFly Core 2.0.10.Final) started in 2564ms
d 267 of 553 services (371 services are lazy, passive or on-demand)
```



Заходим во вкладку Deployments и видим слева список задеплоенных приложений, которые появились из-за того, что их добавила IntelliJ Idea. Для добавления нового нажимаем «add», выбираем «Upload a new deployment».

Выбираем полученный ранее war файл:



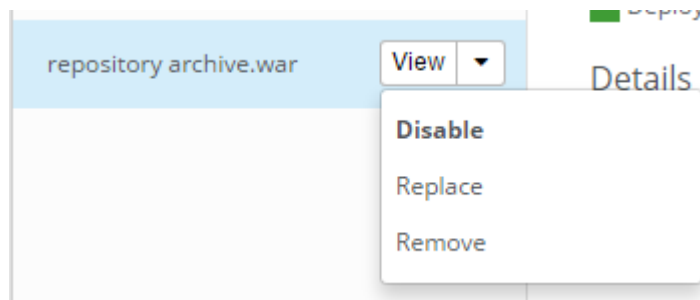
На следующей форме вводим в поле Name уникальное имя, а в поле Runtime Name адрес, по которому будет открываться приложение, например:

The screenshot shows the "New Deployment" dialog box at the "Verify Upload" step. It has a "Need Help?" link in the top right. The "Name \*" field contains "repository archive.war". The "Runtime Name \*" field contains "repo|war". The "Enable \*" checkbox is checked. At the bottom, there is a note: "Required fields are marked with an asterisk (\*)." and three buttons: "Cancel", "« Back", and "Finish".

Нажимаем Finish. Теперь приложение доступно по адресу <http://localhost:8080/repo/>.

## 9.2 Перезагрузка приложения

Перезагрузка может понадобиться, например, когда необходимо поменять настройки в приложении. Для перезагрузки заходим в админку WildFly, переходим на вкладку Deployments. Выбираем нужное приложение и жмем Disable:



Теперь включаем:

