|  |
| --- |
| ИнтерТраст |
| Рабочее место разработчика Платформы ActiveFrame 5 |
|  |

|  |
| --- |
| Митавский Д. В. |

Содержание

[Предусловия 2](#_Toc395881934)

[JetBrains IntelliJ Idea 13 2](#_Toc395881935)

[Вариант простой. Настройка проекта из Git 2](#_Toc395881936)

[Вариант сложный. Настройка проекта вручную. 5](#_Toc395881937)

# Предусловия

Перед настройкой проекта необходимо скачать исходный код из Git-репозитория, настроить Maven-сборку, серверы JBoss и Postgres. В Idea должны быть установлены плагины Maven, GWT, JavaEE, EJB, Spring – выбираются при первоначальной установке или могут быть установлены после неё. Плагины входят в поставку Idea Ultimate Edition.

Корень репозитория Git с проектом Платформы в дальнейшем именуется {ROOT}.

# JetBrains IntelliJ Idea 13

## Вариант простой. Настройка проекта из Git

1. Редактируем .gitignore и комментируем следующие вхождения:

# Idea

#\*.iml

#.idea

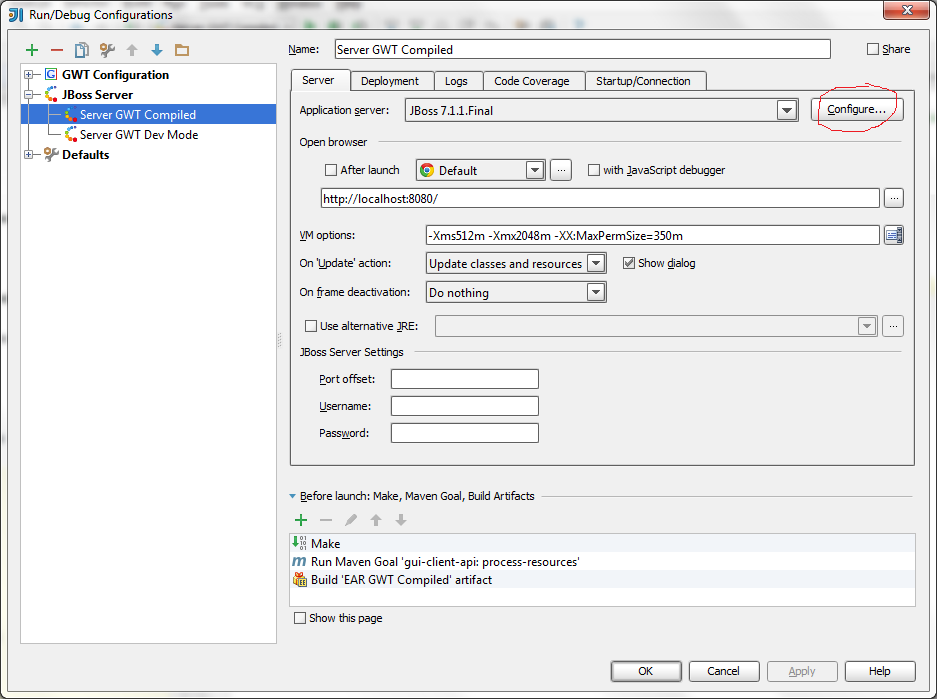
1. Выкачиваем из Git последнюю версию исходников проекта платформы
2. Убираем комментарии в .gitignore:

# Idea

\*.iml

.idea

1. Открываем конфигурацию JBoss и добавляем свой сервер, если этого ещё не сделано:



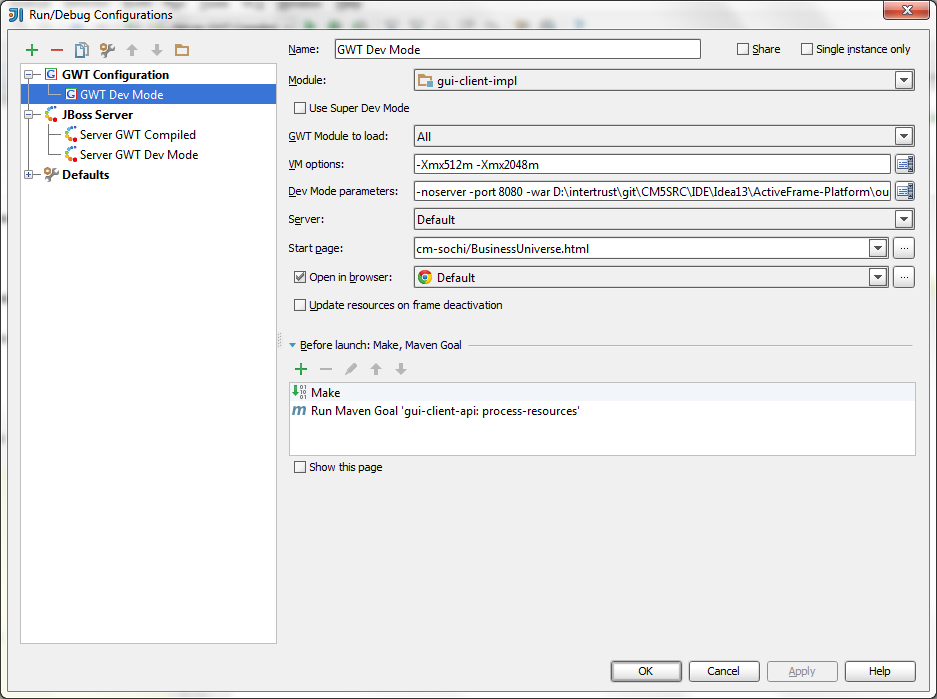
1. Открываем конфигурацию GWT. Устанавливаем Dev Mode parameters:

**noserver -port 8080 -war {DEV\_MODE\_EAR\_ROOT}\web-app.war\**

Нужно заменить **{DEV\_MODE\_EAR\_ROOT}** – это абсолютный путь к месту, где лежит артефакт, предназначенный для работы с GWT Dev Mode. По умолчанию этот путь следующий:

**{ROOT}\IDE\Idea13\ActiveFrame-Platform\out\artifacts\EAR\_GWT\_Dev\_Mode.ear\web-app.war** . В моём случае эти настройки выглядят так:

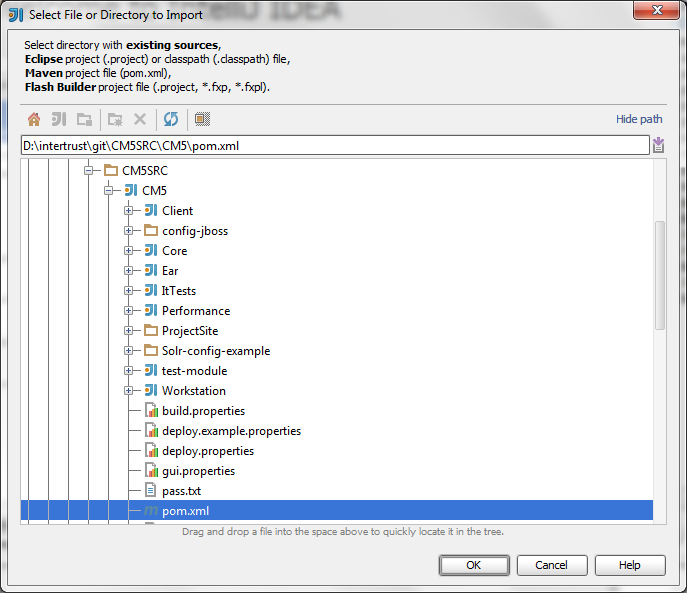
-noserver -port 8080 -war D:\intertrust\git\CM5SRC\IDE\Idea13\1.0-SNAPSHOT\out\artifacts\EAR\_GWT\_Dev\_Mode.ear\web-app.war\



1. Теперь у вас в наличии 3 конфигурации запуска/дебага:
   1. Server GWT Compiled – разворачивается полный EAR с компилированным GWT-кодом
   2. Server GWT Dev Mode – разворачивается EAR без компиляции GWT-кода. Чаще всего, это комбинация используется совместно со следующей:
   3. GWT Dev Mode – запускается режим разработчика GWT. Перед использованием, необходимо запустить сервер приложений (конфигурация Server GWT Dev Mode)
2. Разрабатываем with pleasure!

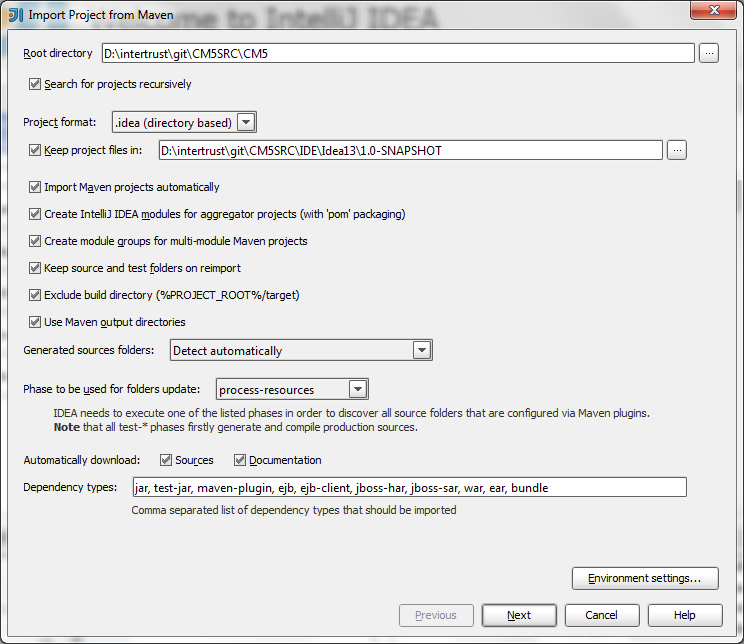
## Вариант сложный. Настройка проекта вручную.

1. На заглавном экране жмём Import Project или File 🡪 Import Project, навигируем к файлу {ROOT}\CM5\pom.xml:



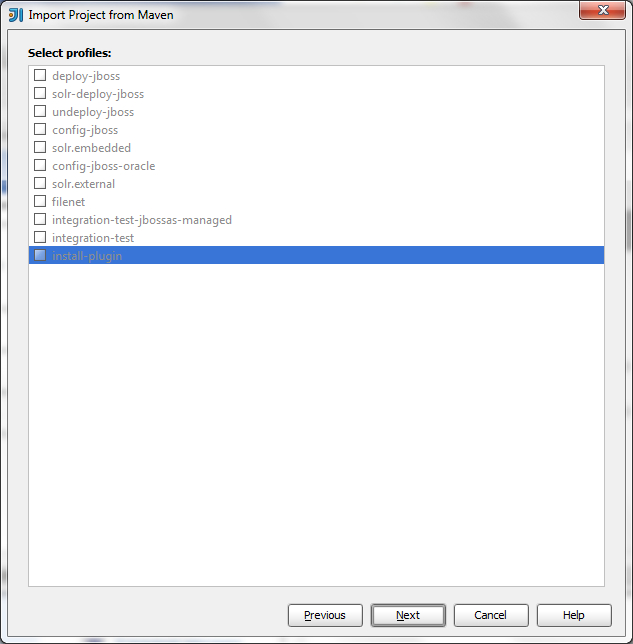
Жмём OK.

1. На следующем экране проставляем абсолютно все доступные галочки. В поле «Keep project files in» указываем каталог хранения файлов проекта. Можно эту галочку снять, но в этом случае Idea будет хранить настройки непосредственно в каталогах исходного кода, что не всегда удобно, и можно случайно-ошибочно опубликовать эти файлы в Git:

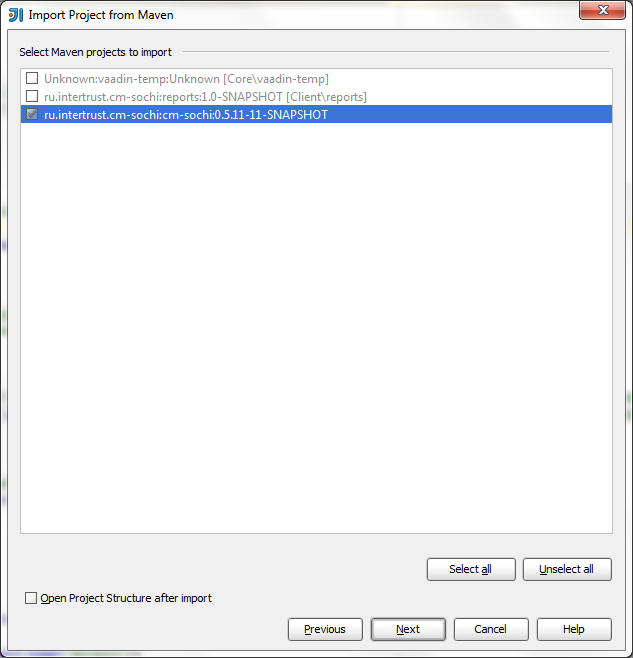


Жмём Next.

1. На следующем экране просто жмём Next, ничего не выбирая:

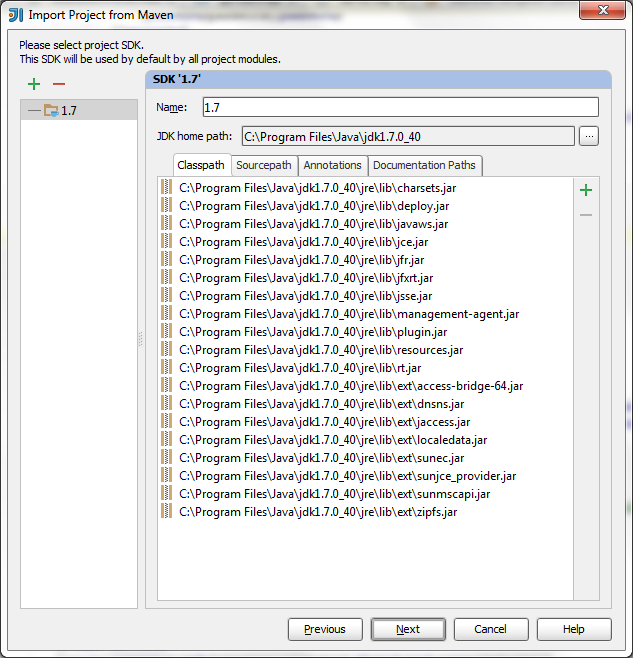


1. На следующем выбираем проект платформы (на момент написания этого документа версии 0.5.11-11-SNAPSHOT):



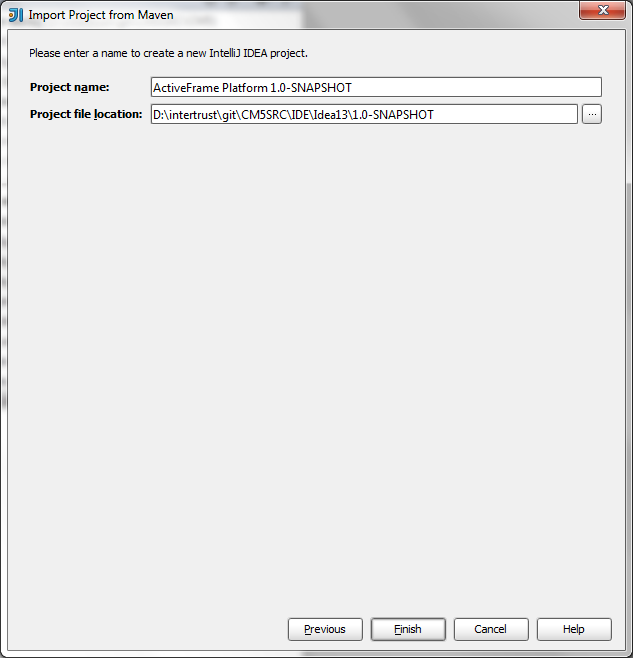
Жмём Next.

1. Выбираем JDK или, если такового нет, добавляем новый:



Жмём Next.

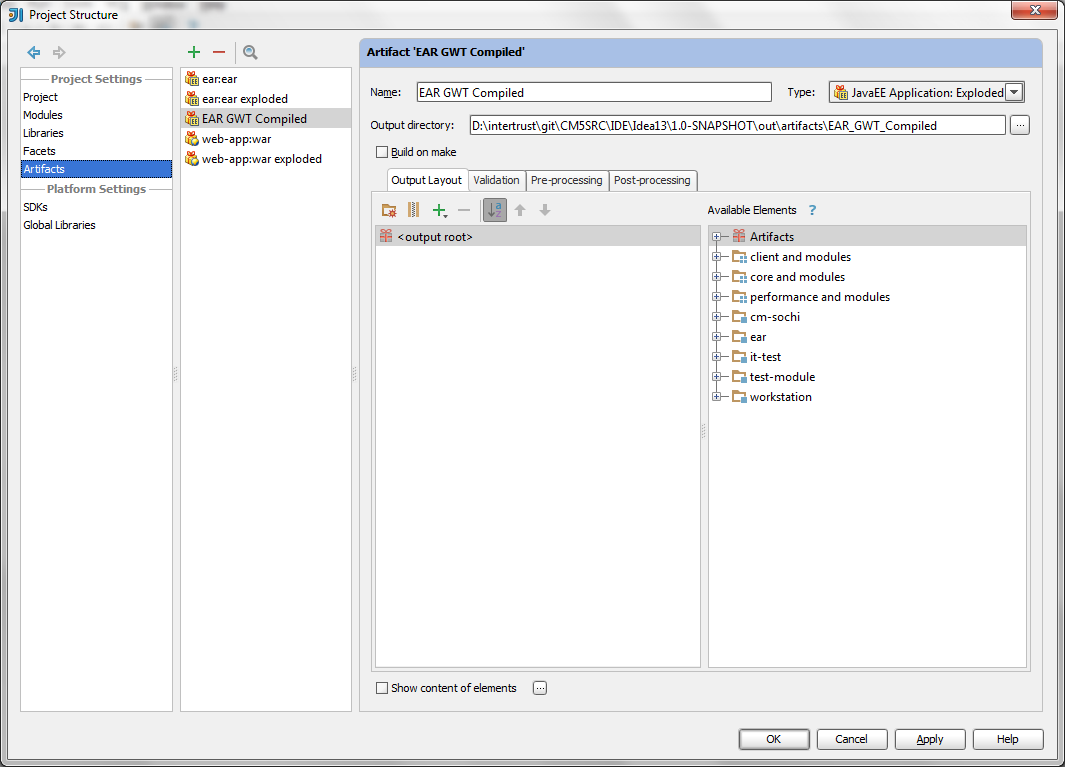
1. Выбираем название проекта. Если он не соответствует умолчанию, то размещения проектного файла при этом собьётся и его, желательно, вернуть, чтобы не разбрасывать проектные файлы Idea по разным папкам:



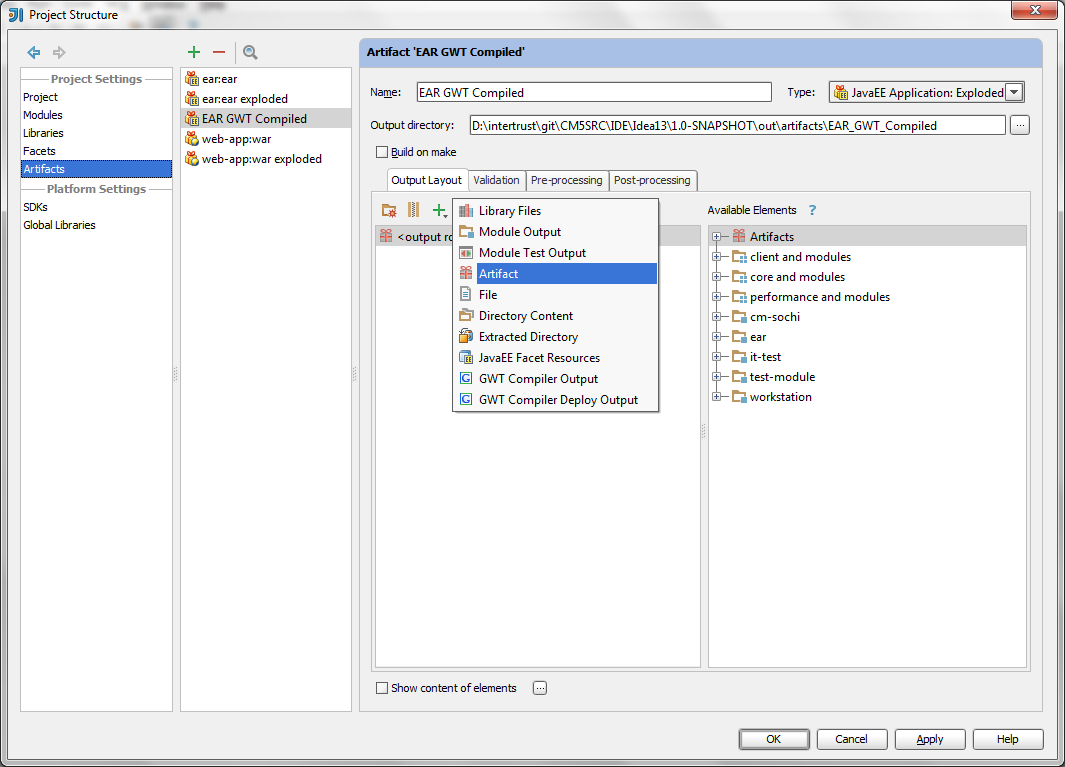
Жмём Finish.

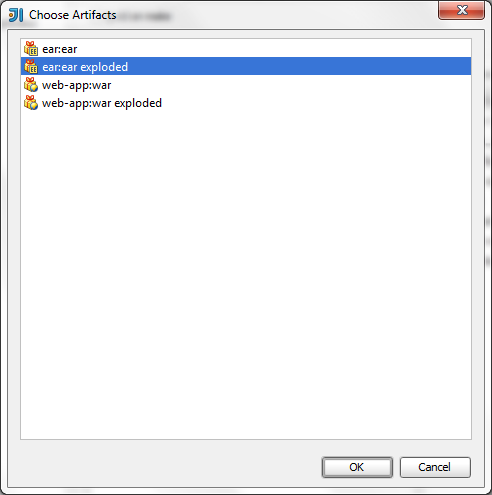
1. После того, как проект откроется, создадим артефакты на основе тех, которые сгенерировала Idea на основе Maven-конфигурации (автоматические получаются несовместимыми с JBoss – то ли из-за специфики наших скриптов, то ли из-за ошибок Idea, а может, из-за всего). Можно было бы модифицировать существующие, но Idea их вернёт в исходный вид, после того, как pom-ы поменяются. Это можно отключить, но тогда после обновления сборочных класс, проект Idea перестанет соответствовать действительности.

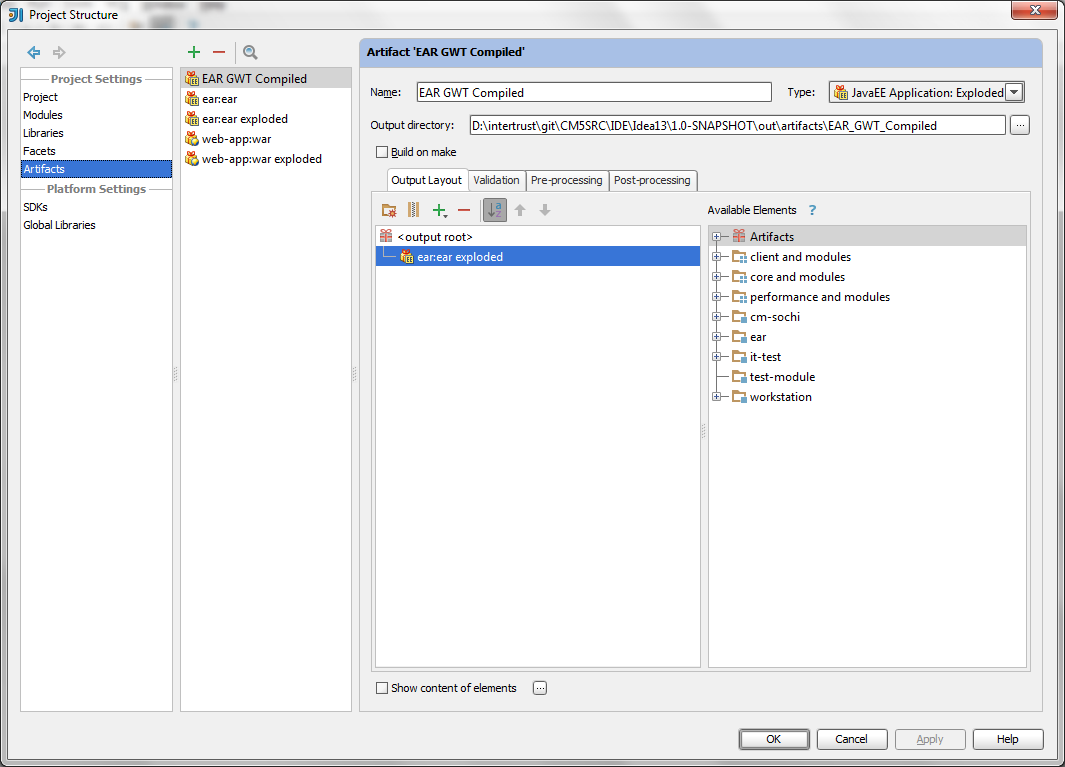
Открываем структуру проекта (File🡪Project Structure🡪Artifacts). Жмём «плюсик» 🡪 JavaEE Application: Exploded. Меняем название:



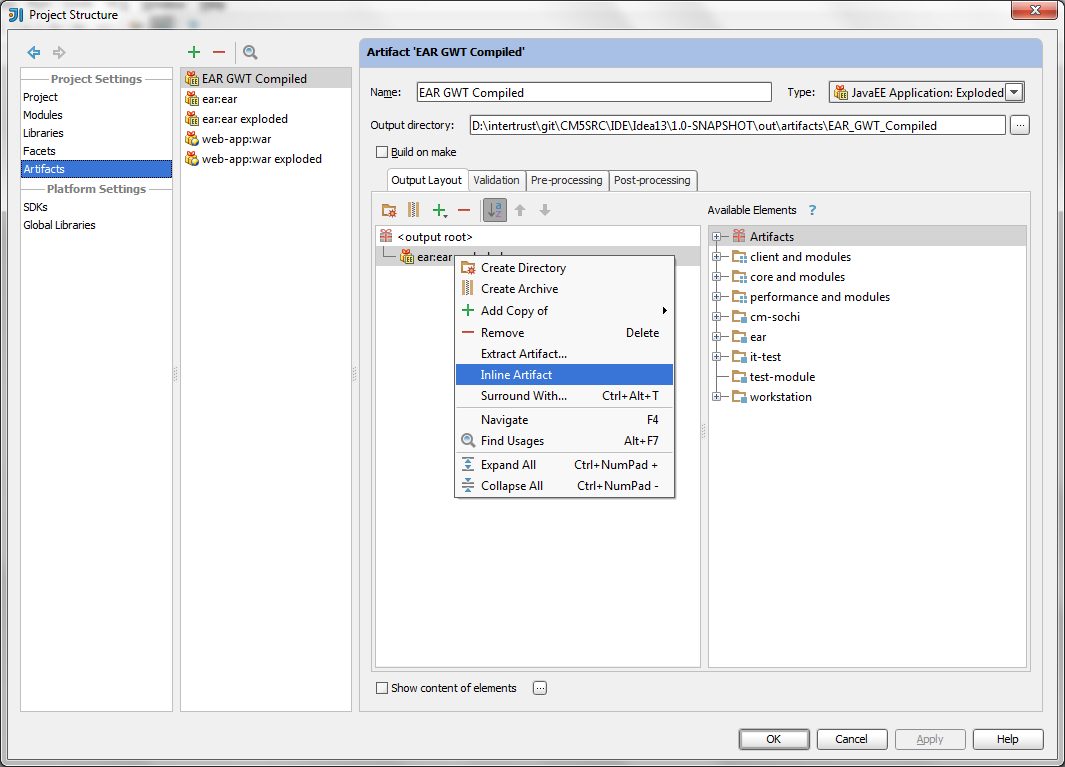
Копируем в наш артефакт содержимое авто-артефакта **ear:ear exploded**: жмём «плюсик» 🡪 Artifact:



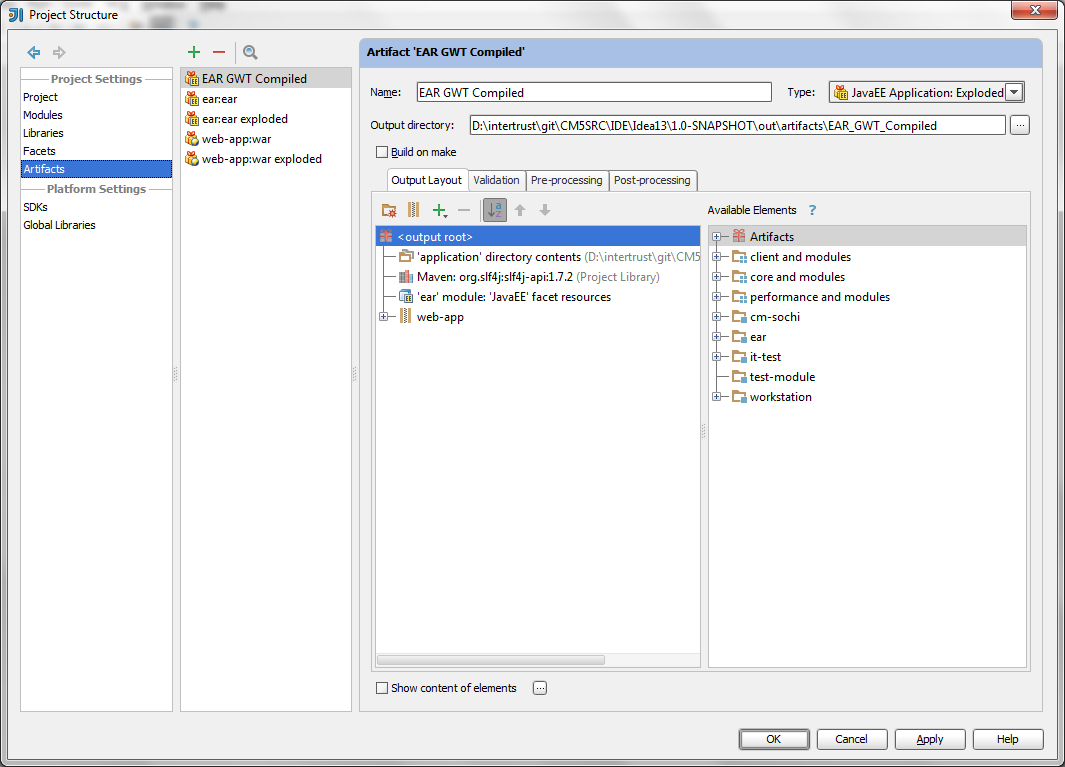




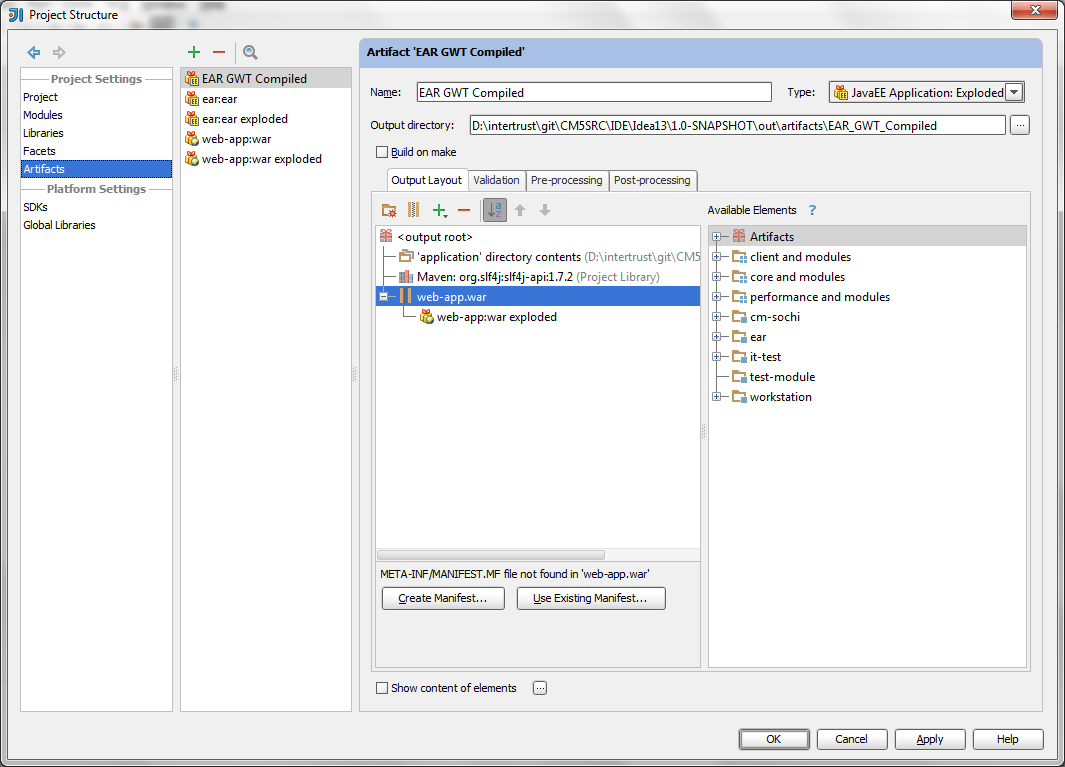
Правой кнопкой мыши жмём в скопированный артефакт внутри корня нового артефакта:



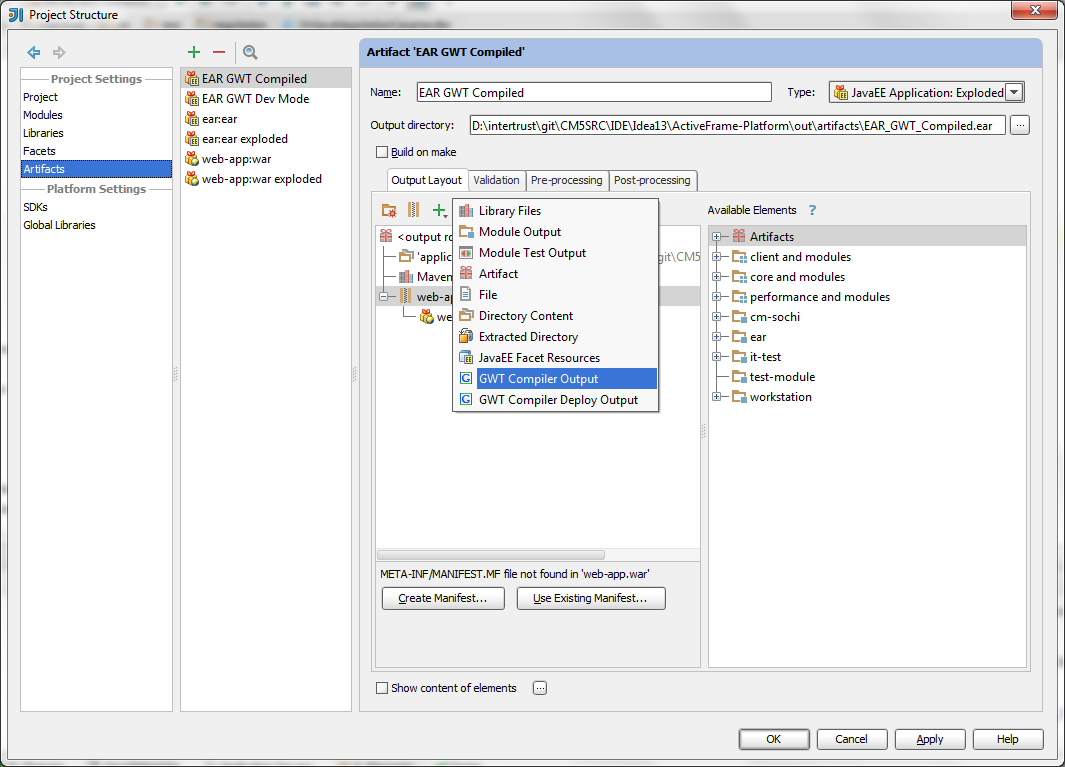
Жмём Inline Artifact.

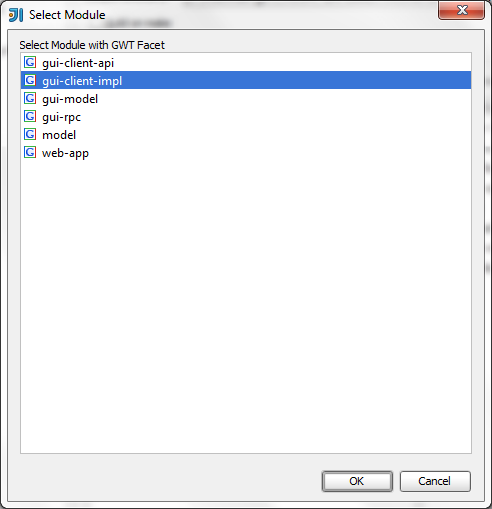


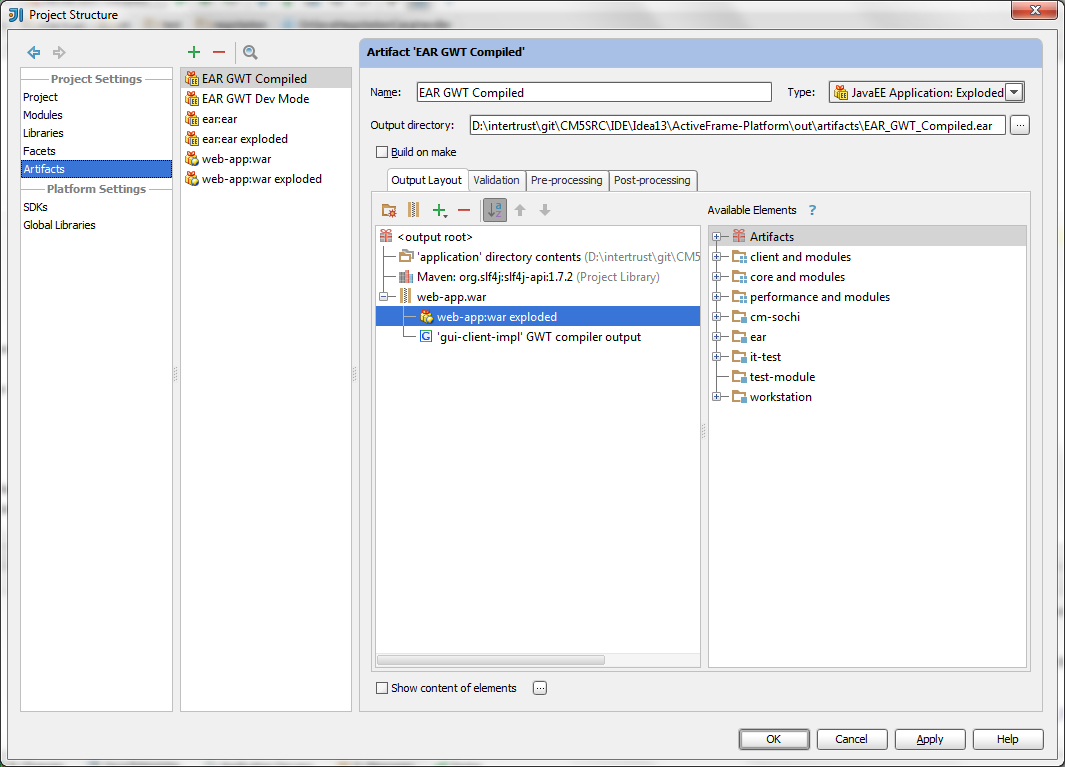
При помощи кнопки «минус» удаляем вхождение **‘ear’ module: ‘JavaEE’ facet resources**, переименовываем папку **web-app** в **web-app.war** (выделяем, жмём Shift+F6):



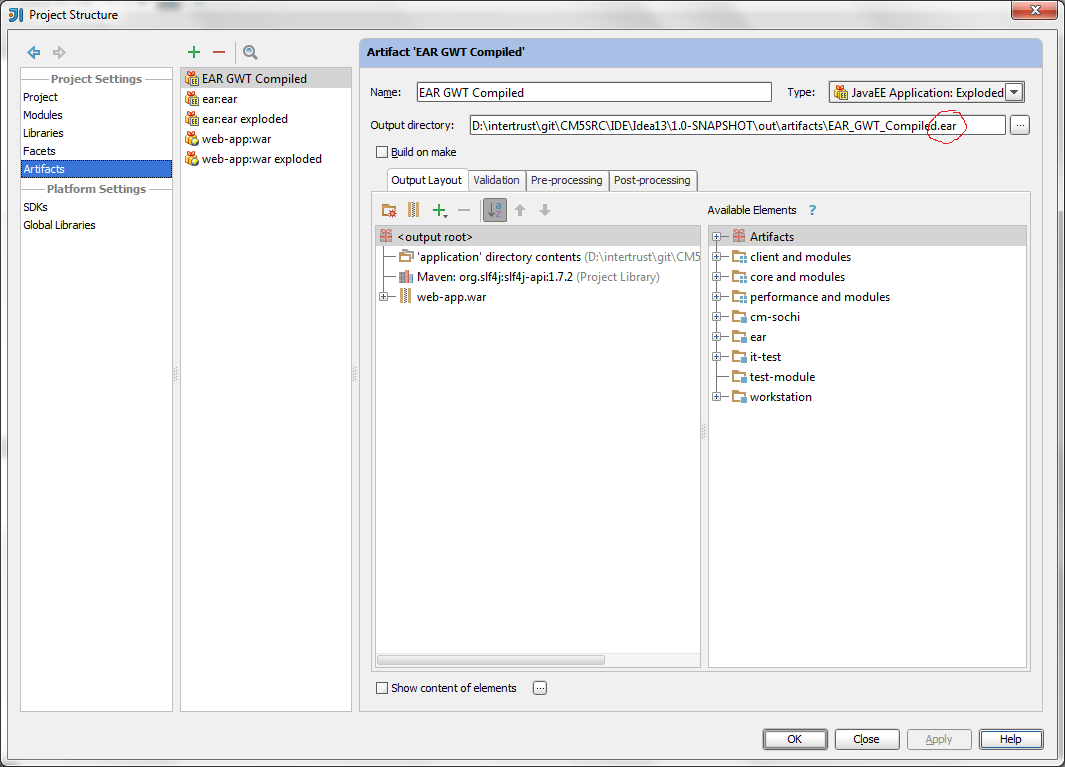
При помощи «плюсика» и Drag’n’Drop добавляем внутрь web-app.war скомпилированнный GWT модуль gui-client-impl:





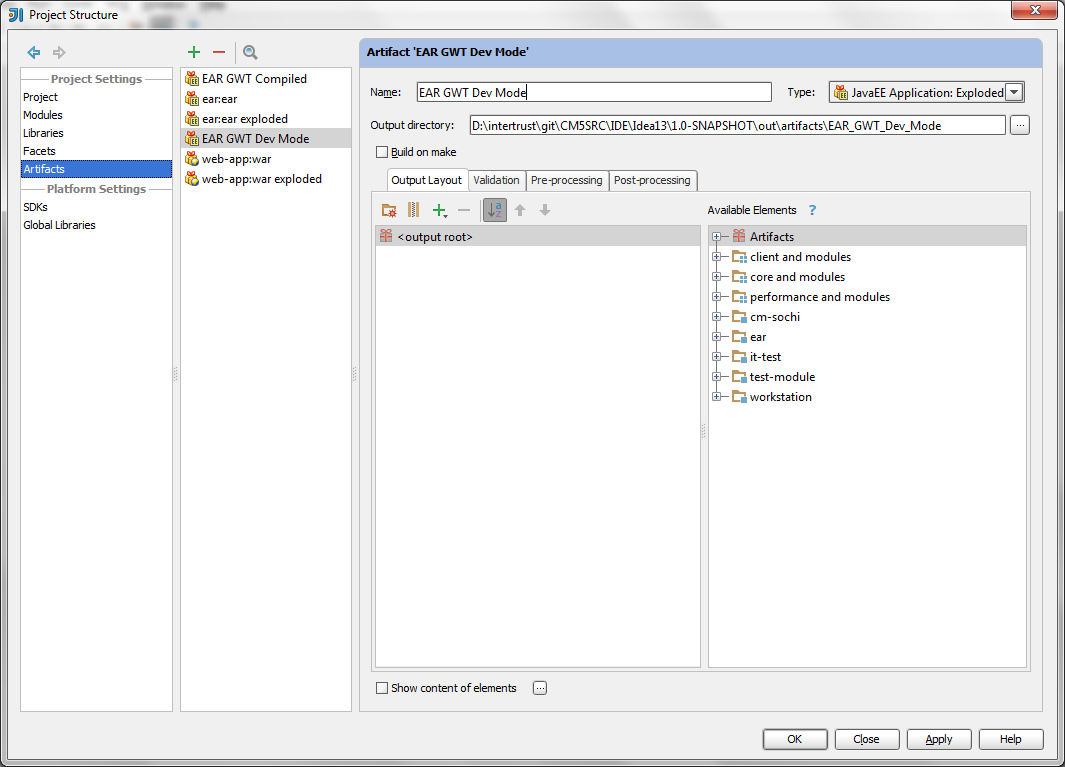


Результирующий католог нашего ear переименовываем, добавляя к нему расширение **.ear** – иначе JBoss не захочет его разворачивать:

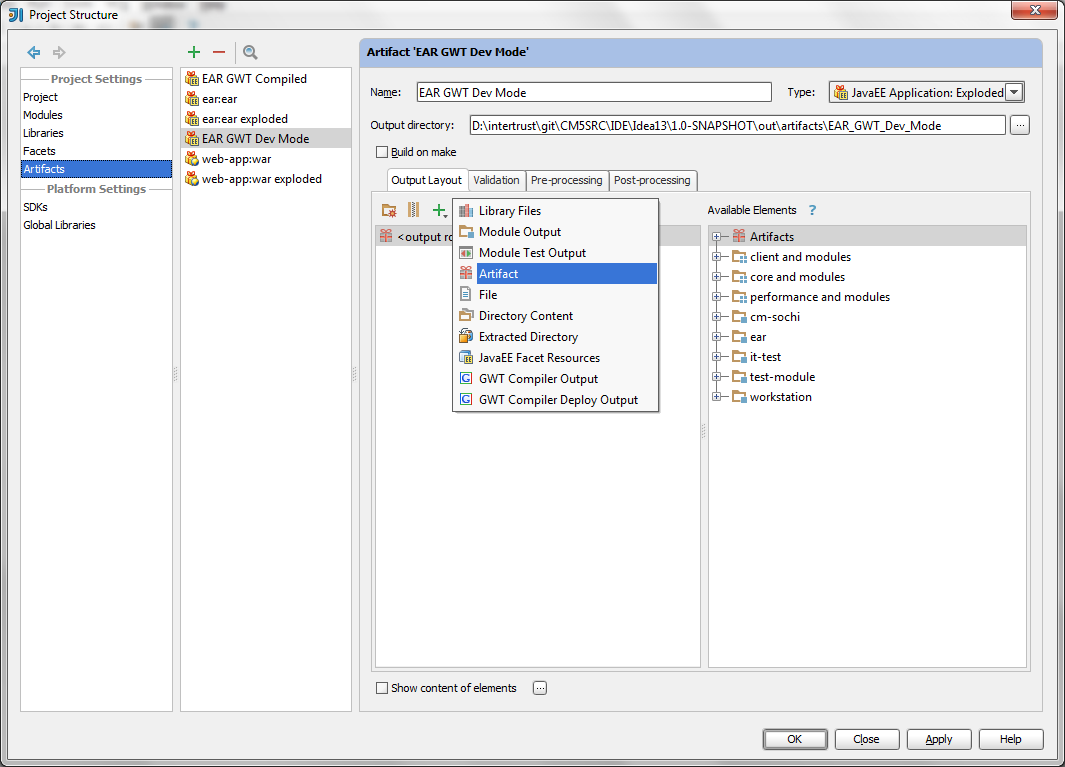


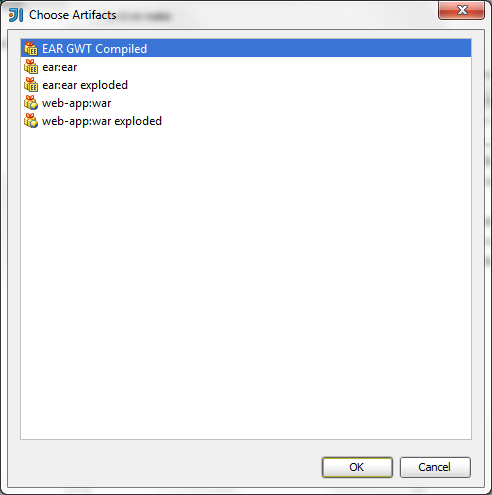
Жмём Apply.

1. Почти аналогичным способом создаём артефакт без GWT-компиляции. Жмём «плюсик» 🡪 **JavaEE Application: Exploded**. Меняем название:

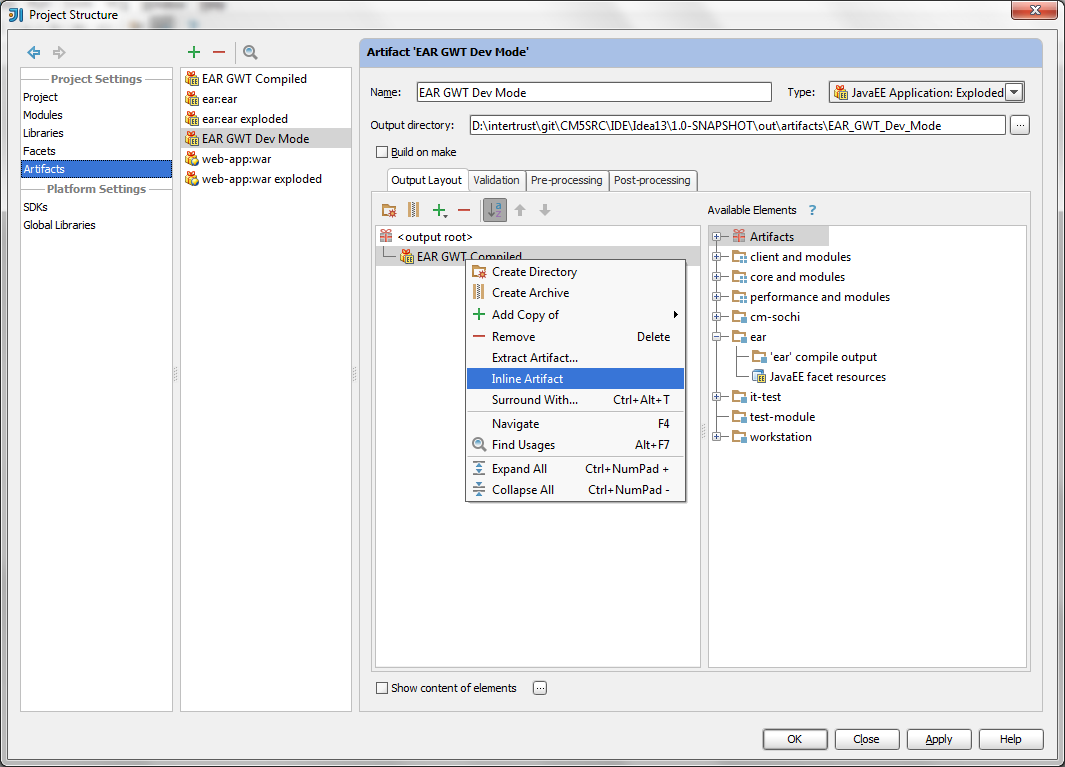


Копируем в наш артефакт содержимое артефакта, только что нами созданного (в нашем примере, **EAR GWT Compiled**):

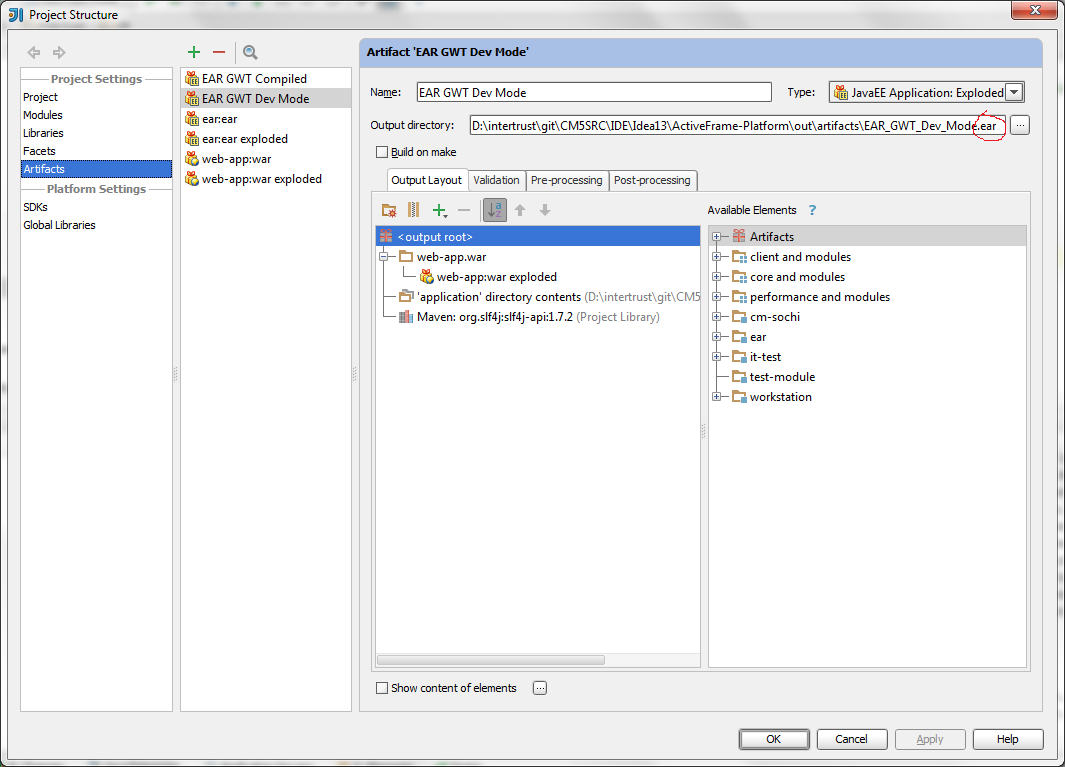




Жмём ОК. Раскрываем содержимое артефакта:

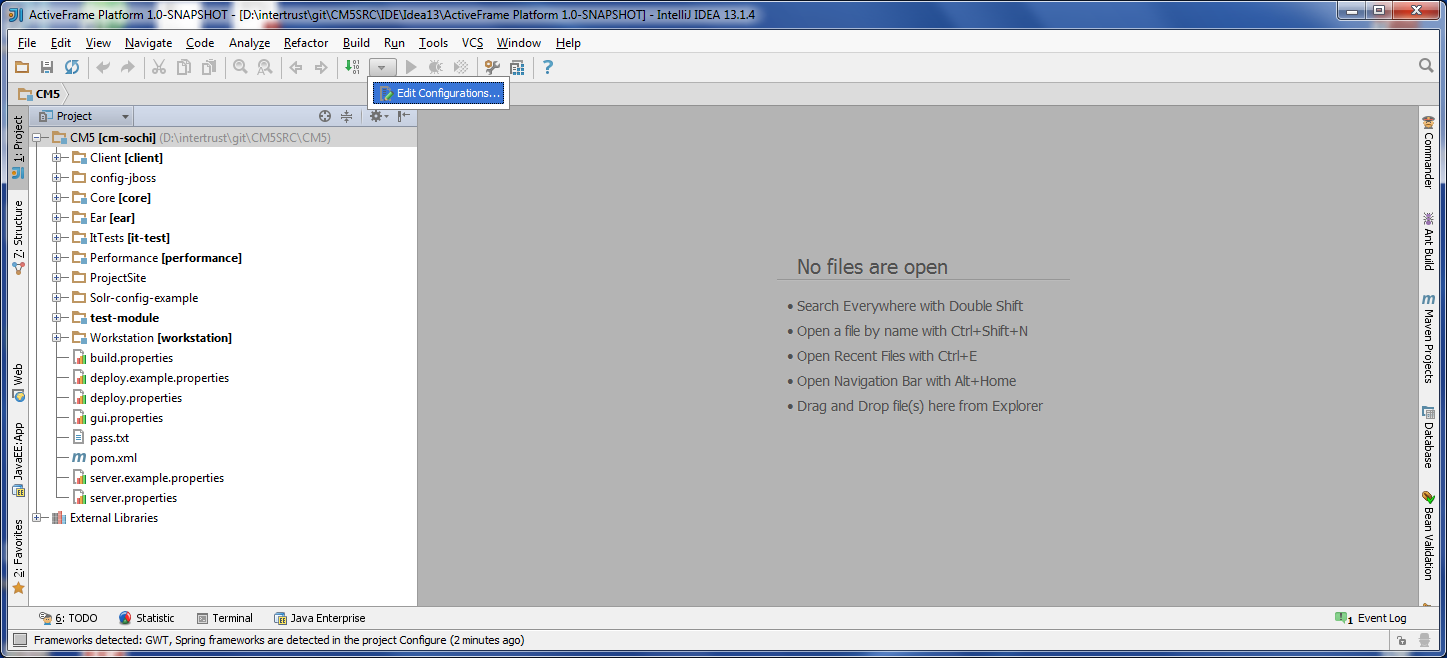


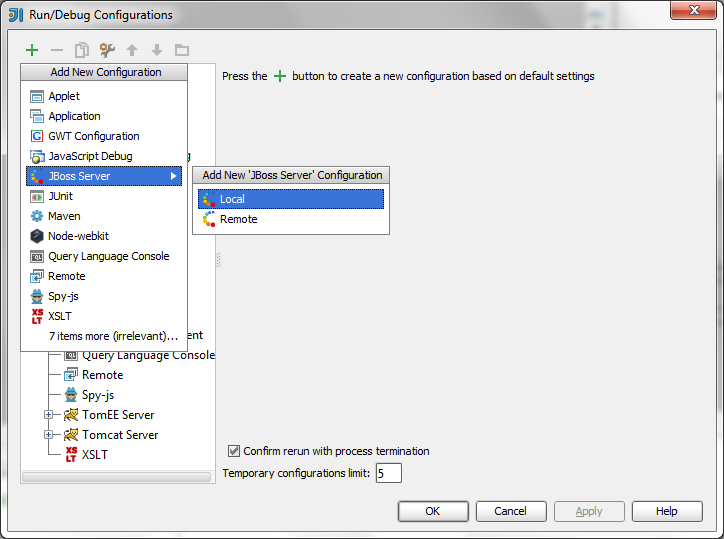
Далее удаляем **архив** web-app.war и добавляем **папку** web-app.war, помещаем в неё артефакт **web-app:war exploded** и переименовываем папку с нашим ear:

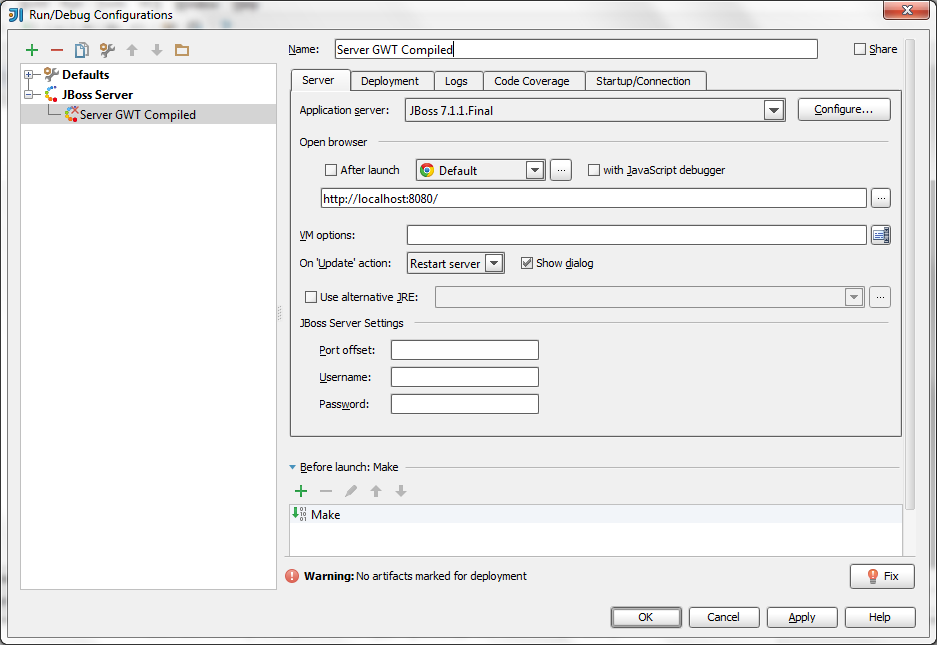


Жмём Apply.

1. Добавляем новую конфигурацию нашего сервера JBoss для запуска при работе в режиме с компилированным GWT кодом. GWT Dev Mode (в этом режиме не будет производиться компиляция GWT):

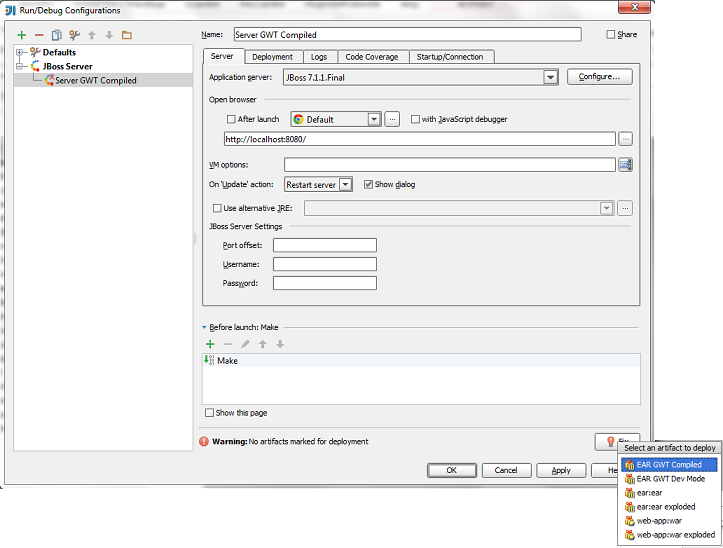


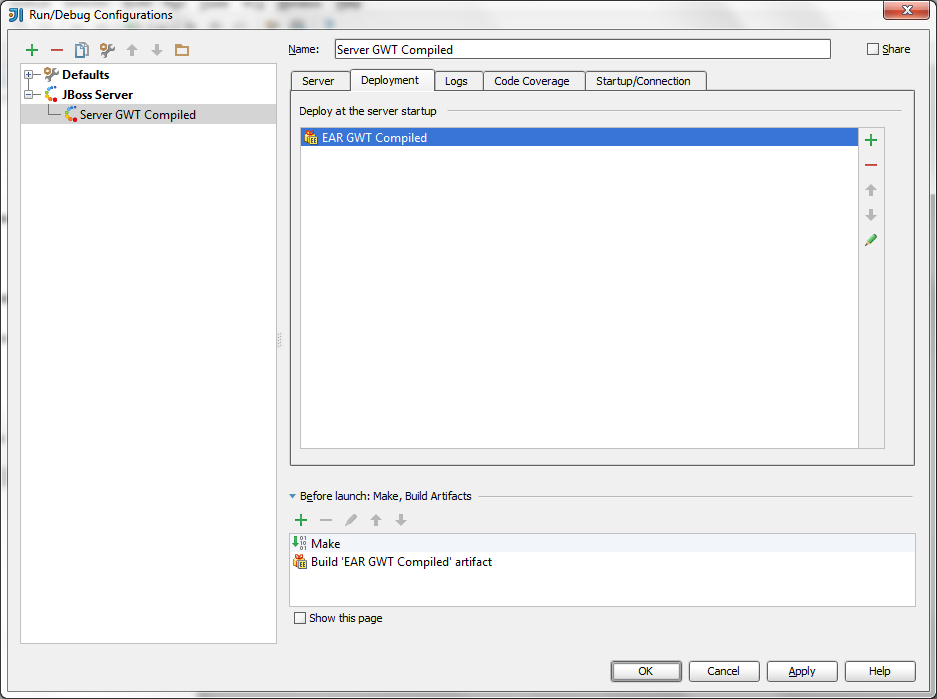




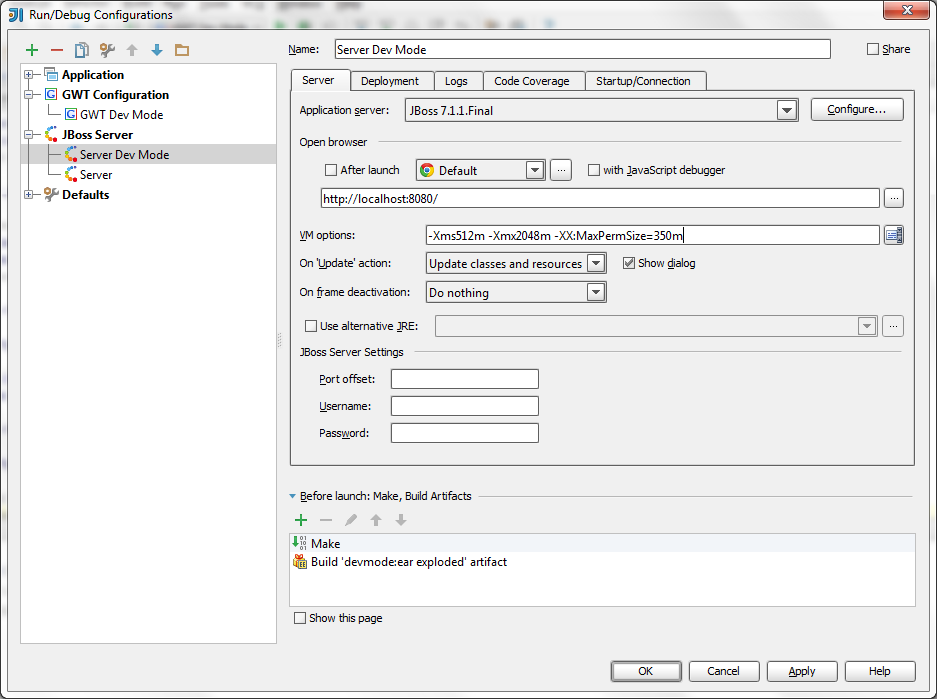
Если Idea не знает ни одного сервера JBoss, жмём кнопку «Configure» и конфигурируем (здесь не приводится описание – это элементарно).

Внизу видим красное предупреждение о том, что ни одни артефакт не разворачивается на этот сервер. Жмём кнопку «Fix», выбираем наш EAR GWT Compiled:



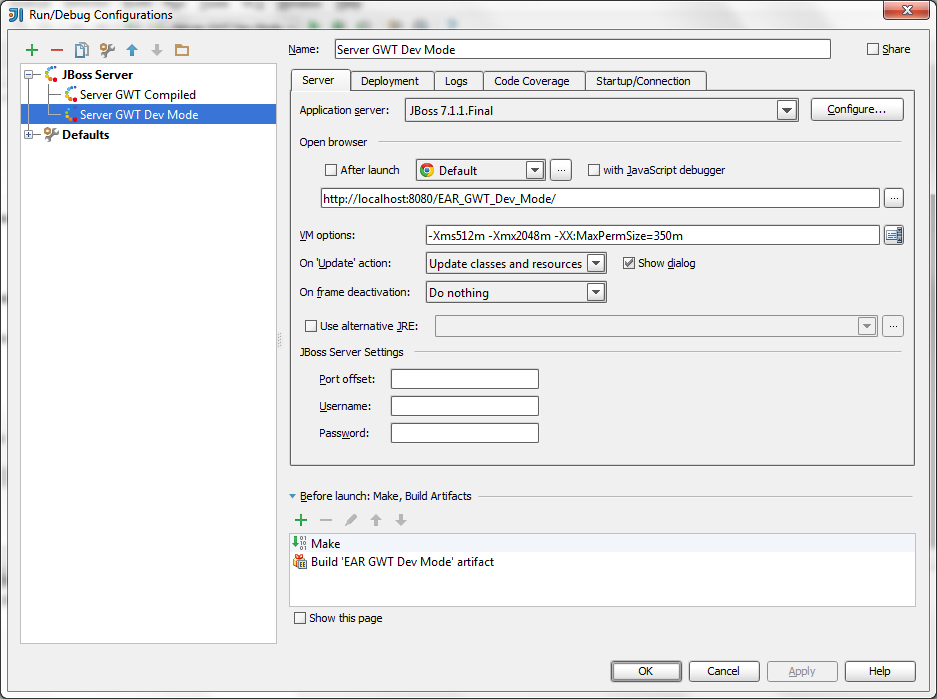


На закладке Server выставляем On ‘Update’ action как «**Update classes and resources**» и прописываем настройки JDK на закладке Server: **-Xms512m -Xmx2048m -XX:MaxPermSize=350m**



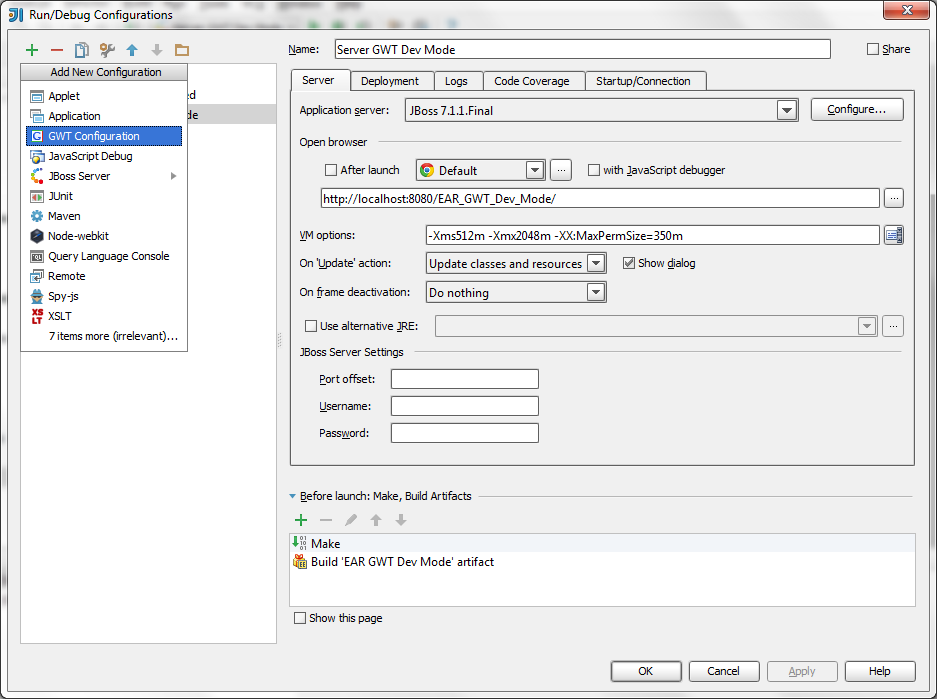
Жмём Apply.

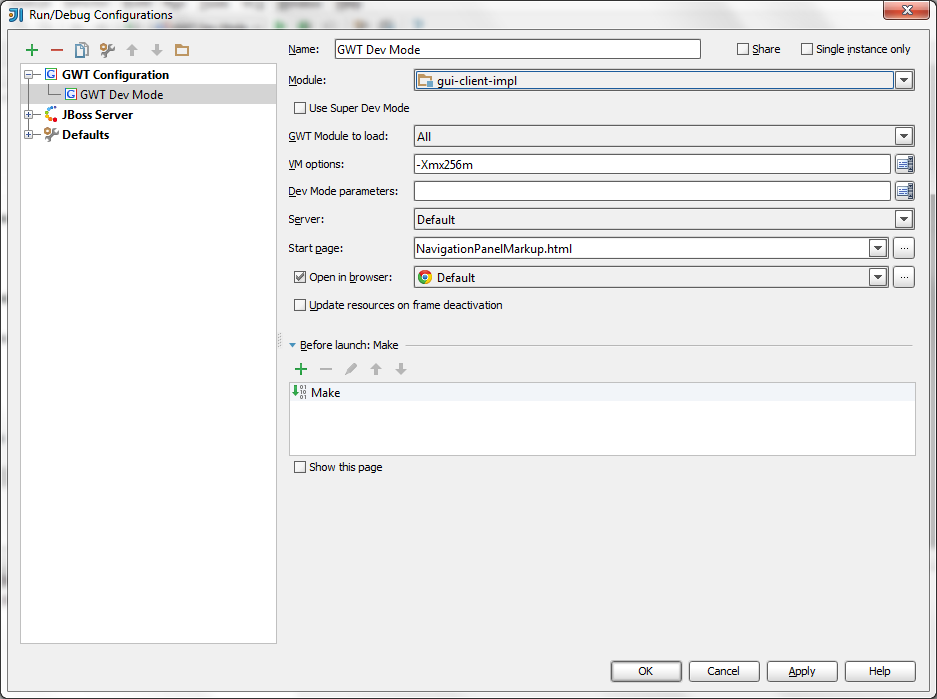
1. Добавляем новую конфигурацию нашего сервера JBoss для запуска при работе в режиме с GWT Dev Mode (в этом режиме не будет производиться компиляция GWT). Это делается аналогично обычному режиму, описанному в пункте 9, нужно только выбрать артефакт EAR GWT Dev Mode и назвать конфигурацию по-другому:



1. Настраиваем GWT Dev Mode:

Добавляем GWT-конфигурацию:





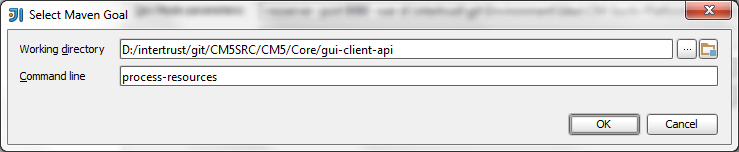
Выбираем модуль gui-client-impl, устанавливаем «пожирнее» опции JDK (я использую **-Xmx512m -Xmx2048m**) , меняем Start page на **cm-sochi/BusinessUniverse.html**, устанавливаем Dev Mode parameters следующим образом:

**-noserver -port 8080 -war {DEV\_MODE\_EAR\_ROOT}\web-app.war\**

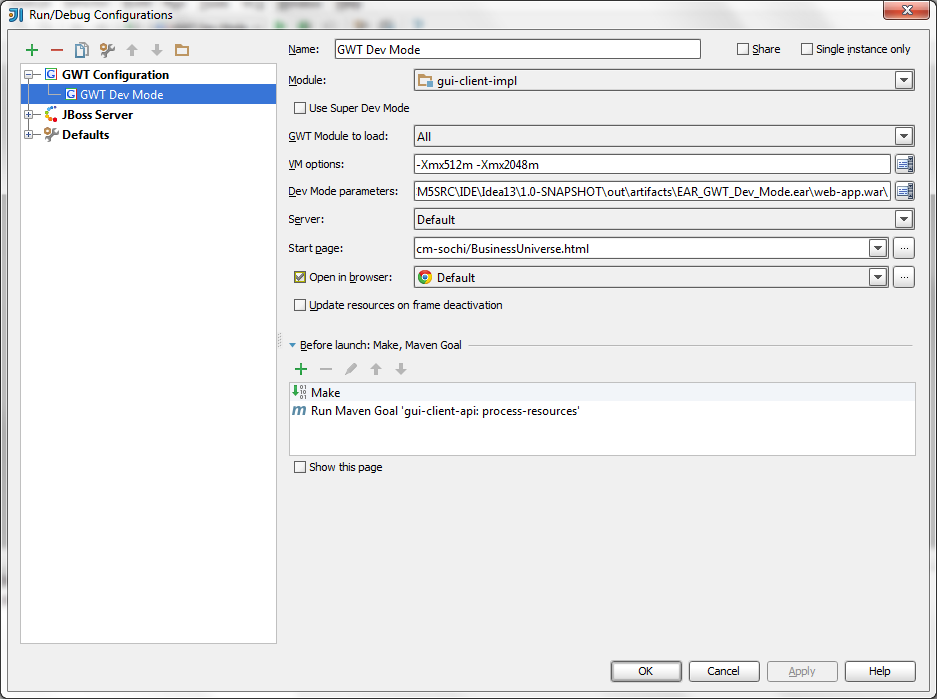
Где **{DEV\_MODE\_EAR\_ROOT}** – это абсолютный путь к месту, где лежит артефакт, предназначенный для работы с GWT Dev Mode. В моём случае эти настройки выглядят так:

-noserver -port 8080 -war D:\intertrust\git\CM5SRC\IDE\Idea13\1.0-SNAPSHOT\out\artifacts\EAR\_GWT\_Dev\_Mode.ear\web-app.war\

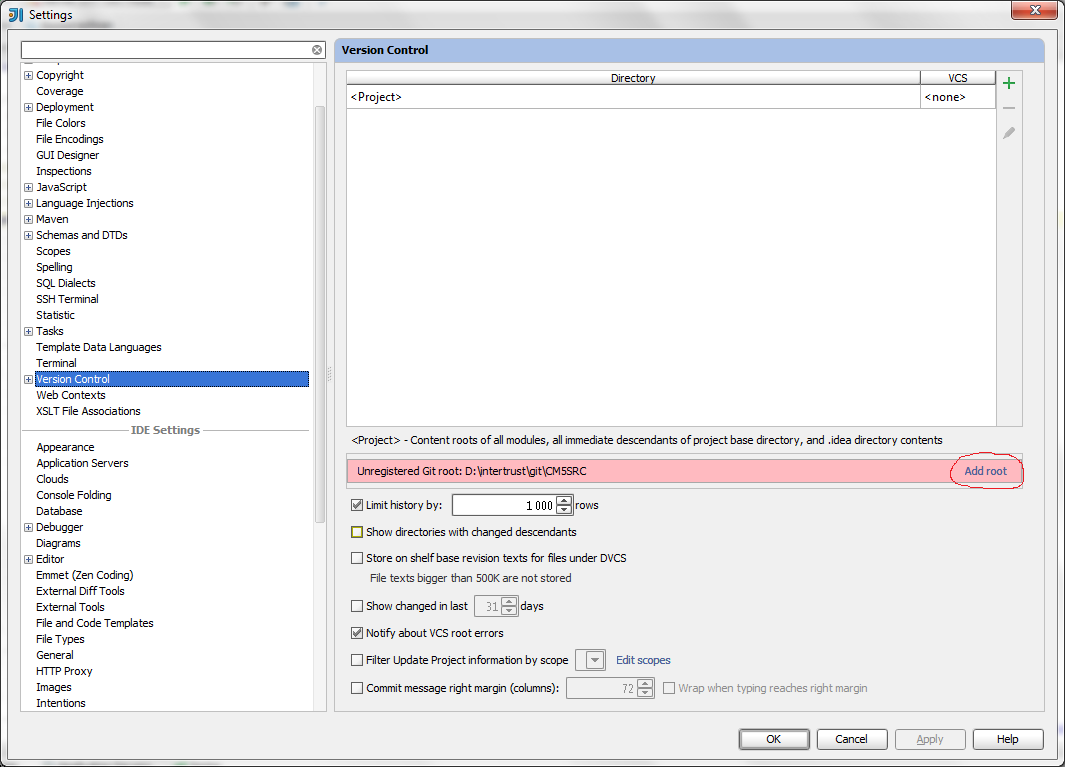
После операцией Make нужно добавить выполнение Maven-цели **gui-client-api:process-resources**. Жмём «плюсик» и выбираем нужную цель:



Жмём ОК:



1. Теперь можно запустить нужную конфигурацию – например Server GWT Dev Mode (сначала) и GWT Dev Mode (после сервера). Выбираем нужную конфигурацию, жмём Run.
2. Для того, чтобы можно было работать с Git из Idea, открываем File 🡪 Settings 🡪 Version Control



Жмём Add root, жмём ОК, разрабатываем with pleasure!