|  |  |
| --- | --- |
|  | Общество с ограниченной ответственностью «Респак» (ООО «Респак») ОГРН 1235400000461  ИНН 5406828500  КПП 540601001  Телефон:+79833210401  Почта: [respak@is-n-s.ru](mailto:respak@is-n-s.ru)  Сайт: respak.team  Адрес: 630007, Новосибирская область, г.Новосибирск, ул. Советская 4а |

**Тестовое задание для кандидата на вакансию Java-разработчика.**

Тестовое задание состоит из двух частей. Первая часть задания обязательна к выполнению, вторая часть может быть выполнена по желанию, но выполненное задание и по второй части будет иметь вес при выборе кандидата для трудоустройства.

**Первая часть тестового задания.**

**Предусловие:**

* Язык программирования: *Java 8;*
* Frameworks: *Spring Boot Starter, Spring Web, Spring JPA, Spring Hibernate;*
* База данных: *Postgres SQL;*

**Ход работы:**

1. Реализовать API для работы с реестром районов
   * Атрибуты районов:
     + название (обязательное)
     + код района
     + статус архивности (да/нет)

Необходимо реализовать следующие запросы:

* Получение списка районов, внесенных в реестр. Реализовать фильтрацию возвращаемого списка по названию и коду района.
* Добавление района
* Изменение записи района
* Отправить в архив (архивные не выводим в реестр)

1. Реализовать реестр фермеров
   * Атрибуты:
     + название организации (обязательное, фильтр)
     + организационно-правовая форма (ЮР, ИП, ФЛ)
     + ИНН (обязательное, фильтр)
     + КПП
     + ОГРН
     + район регистрации (связь с районом/ID - района) (фильтр)
     + районы посевных полей (множественный выбор, связь с районом)
     + дата регистрации (фильтр)
     + статус архивности (да/нет) (фильтр)

Необходимо реализовать следующие запросы:

* Получение списка фермеров, внесенных в реестр. Реализовать фильтрацию возвращаемого списка по указанным атрибутам.
* Получение данных по фермеру. По районам необходимо предоставлять наименования.
* Добавление фермера
* Изменение записи фермера
* Отправить в архив (архивные не выводим в реестр)

*\*один район может содержать несколько фермеров (один ко многим); один фермер может иметь посевные поля в разных районах (многие ко многим)*

**Ожидаемый результат:**

REST-приложение с Open API v3 документацией в виде страницы swagger. Доступ к репозиторию с исходниками приложения. README файл с инструкцией для запуска.

**Вторая часть тестового задания.**

Необходимо разработать фронтенд-приложение “реестр фермеров” для реализованного рaнее api с использованием CMS Liferay версии 6.2

Достаточно будет вывести реестр районов и реестр фермеров (с фильтрацией):

Реализовать реестр районов

* + Атрибуты районов:
    - название
    - код района

Реализовать реестр фермеров

* + Атрибуты:
    - название организации (фильтр)
    - организационно-правовая форма
    - ИНН (фильтр)
    - КПП
    - ОГРН
    - район регистрации (фильтр)
    - районы посевных полей
    - дата регистрации (фильтр)

*примечания:*

*два реестра – два разных портлета (portlet.xml) в рамках одного приложения*

*по атрибутам «фильтр», должна быть возможность фильтрации реестра (*для даты указание диапазона через два поля ввода)

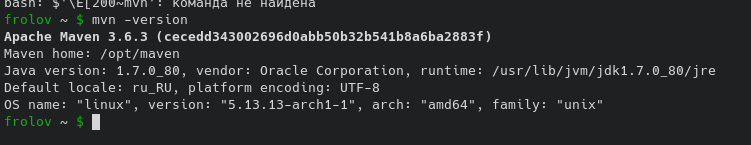
* использовать liferay portal 6.2.5 ga6
* исходный шаблона можно получить чз mvn archetype:generate, название com.liferay.maven.archetypes:liferay-servicebuilder-archetype, информация <https://maven.apache.org/archetype/maven-archetype-plugin/usage.html>
* Вывод информации в виде таблицы можно сделать с использованием компонента <liferay-ui:search-container /> <http://www.liferaysavvy.com/2014/05/liferay-search-container.html>
* Для ввода даты, в лайфрее интегрирован фреймворк AlloyUI, там есть компонент datepicker (<https://alloyui.com/tutorials/datepicker/>)

**Инструкция по установке Liferay 6.2:**

* Установить именно jdk 1.7 (Java SE Development Kit 7u80)<https://www.oracle.com/ru/java/technologies/javase/javase7-archive-downloads.html>
* При невозможности скачать, можно взять из облака <https://disk.yandex.ru/d/RVg_-E5L-KMBQw>
* установить переменную среды JAVA\_HOME и JRE\_HOME
* проверить правильность установки jre командой в cmd : **java -version**
* проверить правильность установки jrk командой в cmd : **javac -version**



* Установить mvn, в переменной PATH указать путь к папке ***{path to maven}/bin***
* проверить правильность установки jre командой в cmd : mvn -version



* Создать системную переменную среды MAVEN\_OPTS со следующим значением: -Xmx512m -Xms256m -Dhttps.protocols=TLSv1.2
* Установить PostgreSQL версию 9.6.23 (<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>)
* Создать базу данных под портал назвать lportal
* Скачать liferay-portal с <https://sourceforge.net/projects/lportal/files/Liferay%20Portal/6.2.5%20GA6/liferay-portal-tomcat-6.2-ce-ga6-20160112152609836.zip/download>
* Запустить портал через [liferay-home]\tomcat-7.0.62\bin\startup.bat
* После открытия стартовой страницы, указать первоначальную конфигурацию БД, нажав на **Change**. Выбрав postgresql, указав логин postgres, пароль который при установки базы данных. Остальные параметры можно оставить по умолчанию
* Можно создавать первый портлет. Для этого используем maven архетипы в cmd:

**mvn archetype:generate -Dfilter=com.liferay.maven.archetypes:**

Команда покажет список доступных архетипов liferay приложений

Выбираем liferay-portlet-archetype

Выбираем версию портала 6.2.5

Задаем группу и название, остальное можно скипать

В рабочей директории будет создана папка с шаблоном

в pom.xml в проекте нужно указать код ниже, и только изменить путь до liferay!

**<properties>**

**<compiler.maven.source>1.7</compiler.maven.source>**

**<compiler.maven.source>1.7</compiler.maven.source>**

**<liferay.version>6.2.5</liferay.version>**

**<liferayVersion>${liferay.version}</liferayVersion>**

**<liferay.maven.plugin.version>${liferay.version}</liferay.maven.plugin.version>**

**<portal.dir>D:\\liferay-portal-6.2-ce-ga6</portal.dir> *← здесь и удалить этот комментарий***

**<tomcat.dir>${portal.dir}/tomcat-7.0.62</tomcat.dir>**

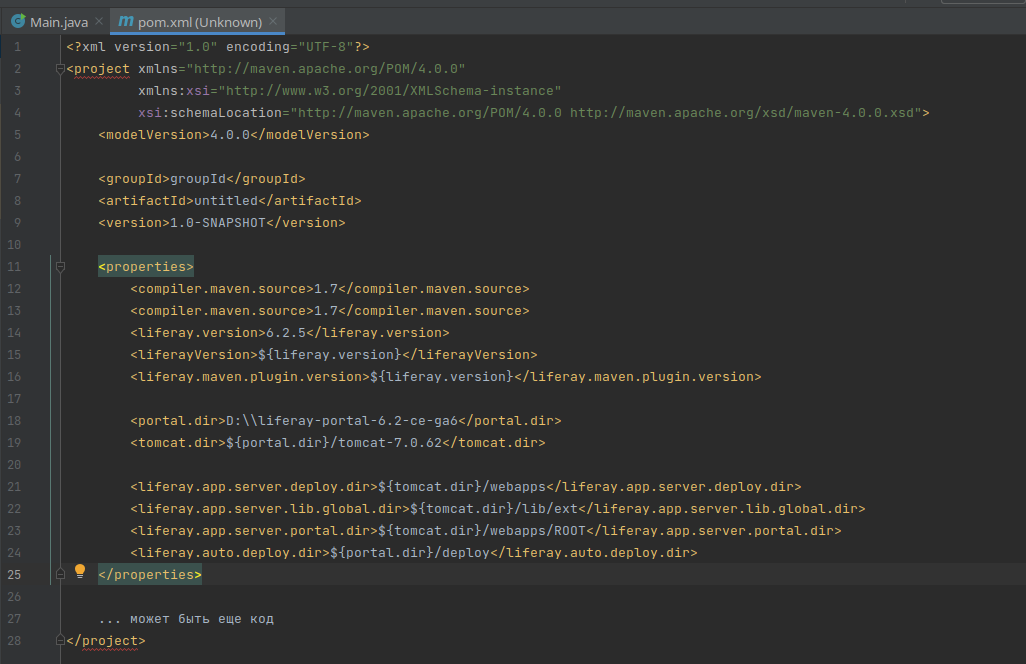
**<liferay.app.server.deploy.dir>${tomcat.dir}/webapps</liferay.app.server.deploy.dir>**

**<liferay.app.server.lib.global.dir>${tomcat.dir}/lib/ext</liferay.app.server.lib.global.dir>**

**<liferay.app.server.portal.dir>${tomcat.dir}/webapps/ROOT</liferay.app.server.portal.dir>**

**<liferay.auto.deploy.dir>${portal.dir}/deploy</liferay.auto.deploy.dir>**

**</properties>**

  
Длее можно приступать к указанию портлетов в  
**src\main\webapp\WEB-INF  
cозданию jsp в  
src\main\webapp\jsp**  
Контролеров и прочей java соотв в  
**src\main\java**

* Для сборки проекта используем mvn clean package
* есть 3 способа загрузить war архив на сервер
  + 1) через сам браузер, выбераем панель управление → Менеджер приложений→ установить→ там выбираем файл и нажимает кнопку установить
  + 2) через папку. копируем собранный war архив из папки **target** проектав папку **deploy** в [liferay-home]
  + 3) через консоль в idea вводя команду **mvn liferay:deploy**

Уроки для ознакомелния с работой в liferay весь код приведен в самом низу страницы→

урок как работать→

https://help.liferay.com/hc/en-us/articles/360018176951-Writing-Your-First-Liferay-Application

Доп. Материал, где подробно описаны шаги по созданию проекта

https://liferayiseasy.blogspot.com/2015/02/creating-portlet-and-services-using.html

http://www.liferaysavvy.com/2014/05/liferay-mvc-portlet-form-submission.html

http://www.liferaysavvy.com/2014/05/liferay-mvc-portlet-database.html