**IT7374 Programming III - Assignment 2**

**Distributed Application in Java**

**Project Name:** CRUD\_Example

CRUD\_Example\_Web\_Service

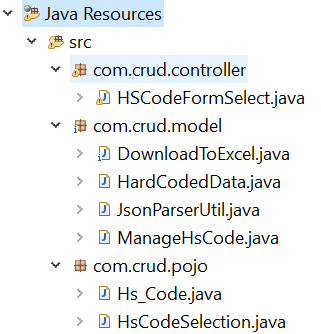
**Brief Description:**

1. **CRUD\_Example –** The project is all about accessing (display or download to excel file) and managing HS\_Code DB created in Oracle. Managing HS\_Code DB means update, delete and creation of a new record.
2. **CRUD\_Example\_Web\_Service –** This is the web service used in downloading HS\_Code to excel file. Though I know that I can create web services inside CRUD\_Example project, however, in this CRUD\_Example\_Web\_Service I have shown different approach like using HQL (Hibernate Query Language) to access Oracle DB and creating a stand-alone Jetty server.

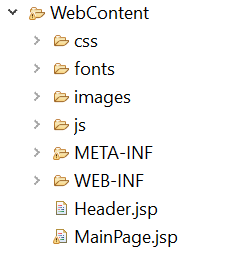
**Detailed Project Specifications:**

1. **CRUD\_Example**

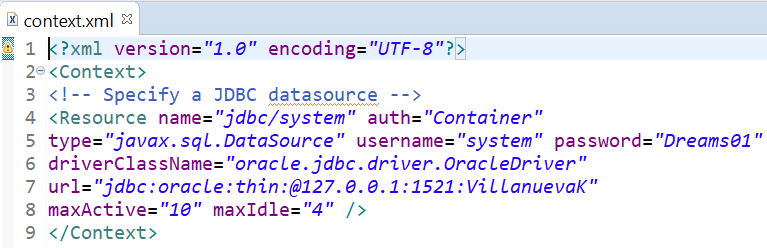
* The following technologies/API were used to build this project:
* Java Servlets and Java Server Pages (JSP)
* JSP Standard Tag Library (JSTL)
* Java Database Connectivity (JDBC)/Database Programming
* Oracle 11g Database
* Apache Tomcat Server 9.0
* Bootstrap/JQuery/JS – for the design
* XML for storing Oracle DB Connection hardcoded values i.e. Username/Password/Server Name and URL
* POI – for excel file creation
* Jersey and GSON to parse JSON Array which is the result of the web service coming from CRUD\_Example\_Web\_Service
* File I/O and Object Serialization
* Internationalization for Thai Characters
* In ***Java Resources*** folder, I created 3 packages named as:
* com.crud.controller – this package will hold all the servlets.
* com.crud.model – the package will house in all java classes except from servlets.
* com.crud.pojo – this will keep all the DB classes or Pojo



* ***WebContent*** folder of the same project, this is where all the CSS/JS/JQuery and JSP Pages can be found.

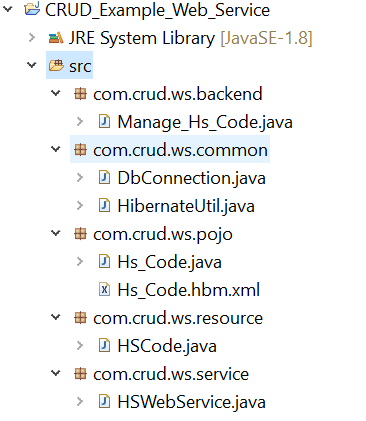


* In ***WebContent 🡪 Meta-INF*,** I created context.xml to store the Oracle DB server name, URL, username and password as follows:

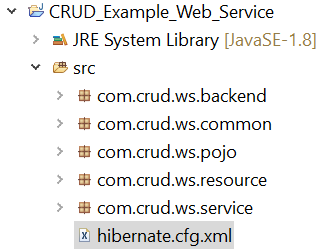
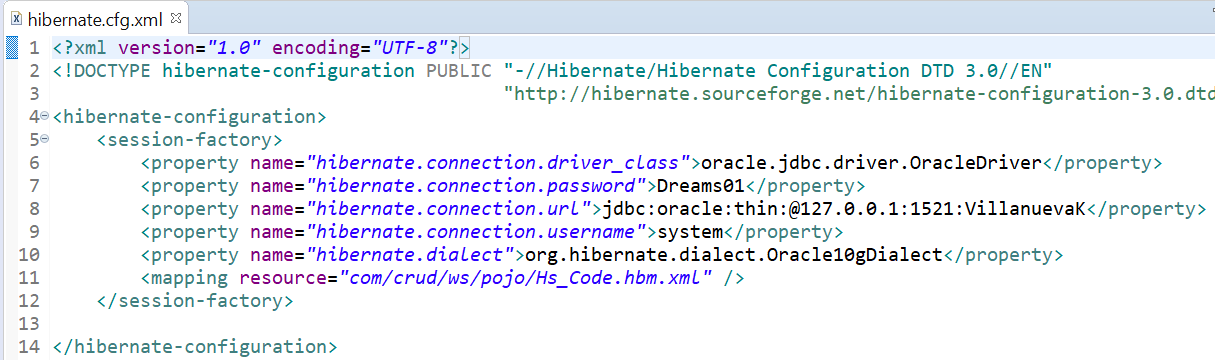


1. **CRUD\_Example\_Web\_Service**

* The following technologies/API were used to build this project:
* XML/HQL (Hibernate Query Language) and JBOSS to access Oracle DB
* Jersey/Jetty – to create a stand-alone server
* JDBC and Servlet
* GSON – to return JSON String
* Advanced Rest Client – to test the created web service
* In ***src*** folder, I created the following packages:
* com.crud.ws.backend – this is where to keep all the Java classes accessing Oracle DB
* com.crud.ws.common – common classes that can be used by any of the Java classes
* com.crud.ws.pojo - this will keep all the POJOs and Hibernate XML
* com.crud.ws.resource – the package will house-in all the RESTful service resources
* com.crud.ws.service – this is the package where I keep the Java class that will start the server

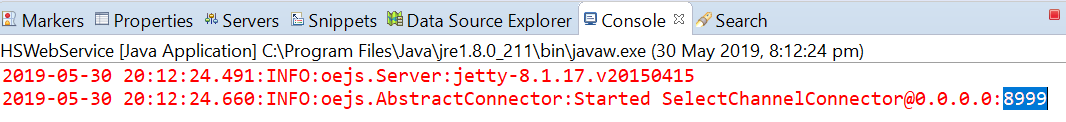


* In ***src*** folder again, hibernate.cfg.xml is created, this the configuration file where you can locate the Oracle DB settings used in accessing it, i.e. user/password/URL and Oracle dialect.

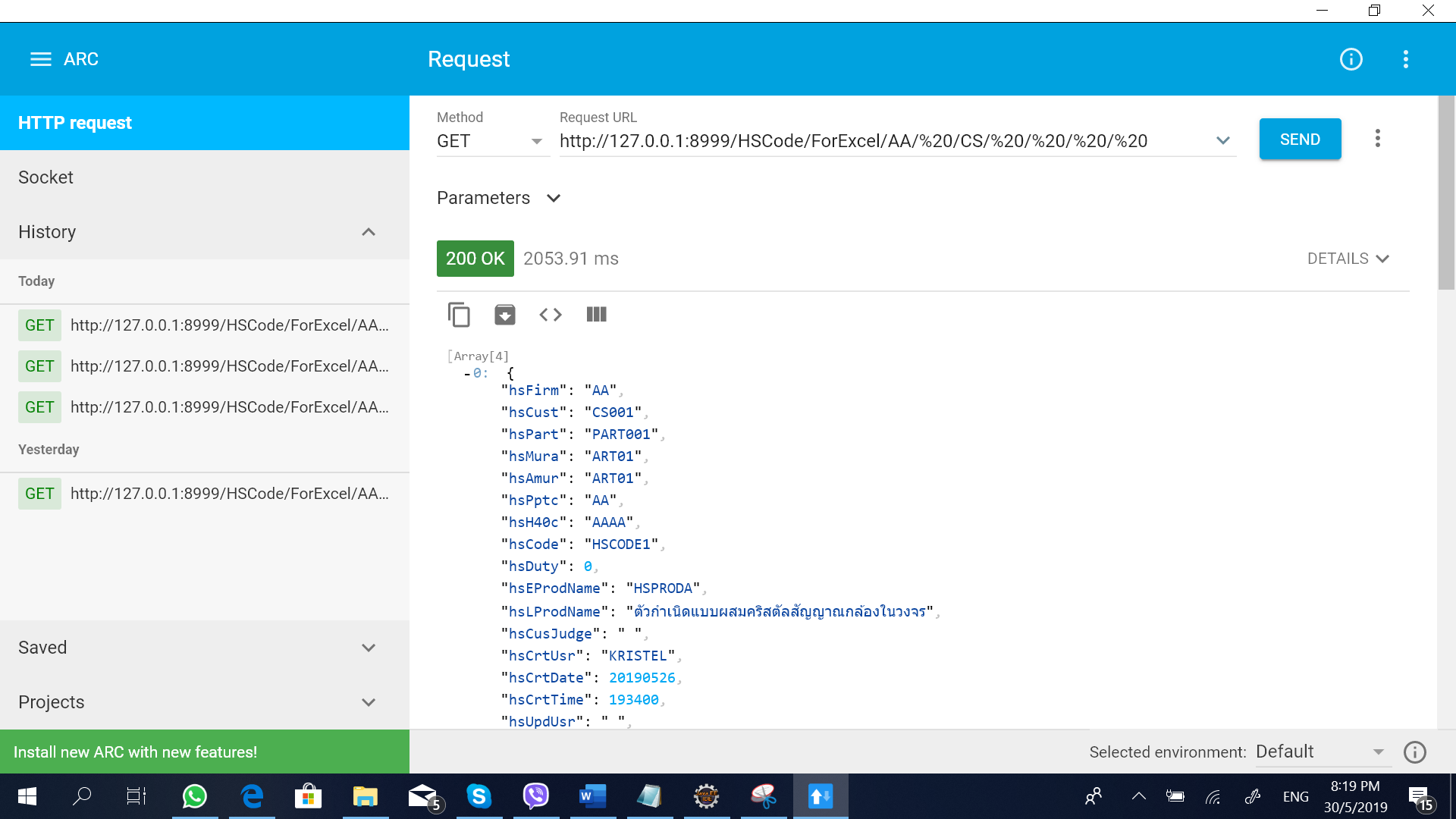
**How to start the HS Code Maintenance Application and Sample Screen shot showing the App’s testing result.**

1. Start the Web Service first under ***CRUD\_Example\_Web\_Service*** by running *com.crud.ws.service/HSWebService.java* as *Java Application*. Once it was successfully started you can see the below message in console.

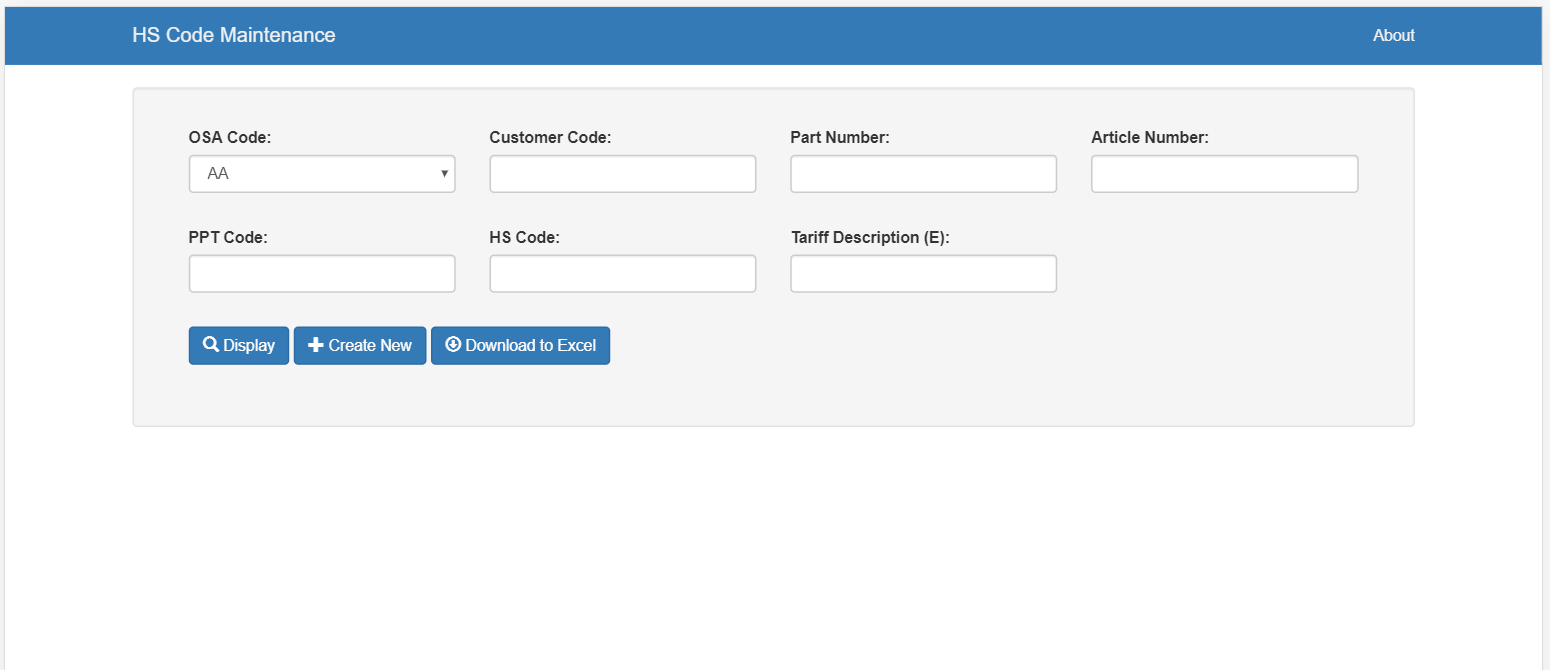


***Note:*** The port no. 8999 is hard coded in the program, however, in actual production this should be passed as parameter.

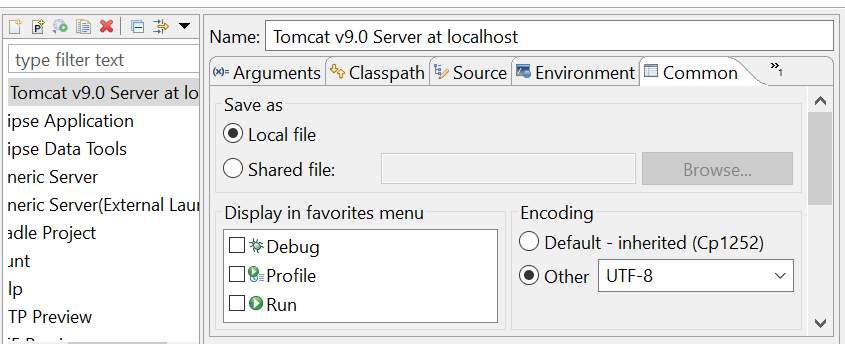
1. Test if the web service is working by using Advanced Rest Client and this Request URL 🡪 <http://127.0.0.1:8999/HSCode/ForExcel/AA/%20/CS/%20/%20/%20/%20>.



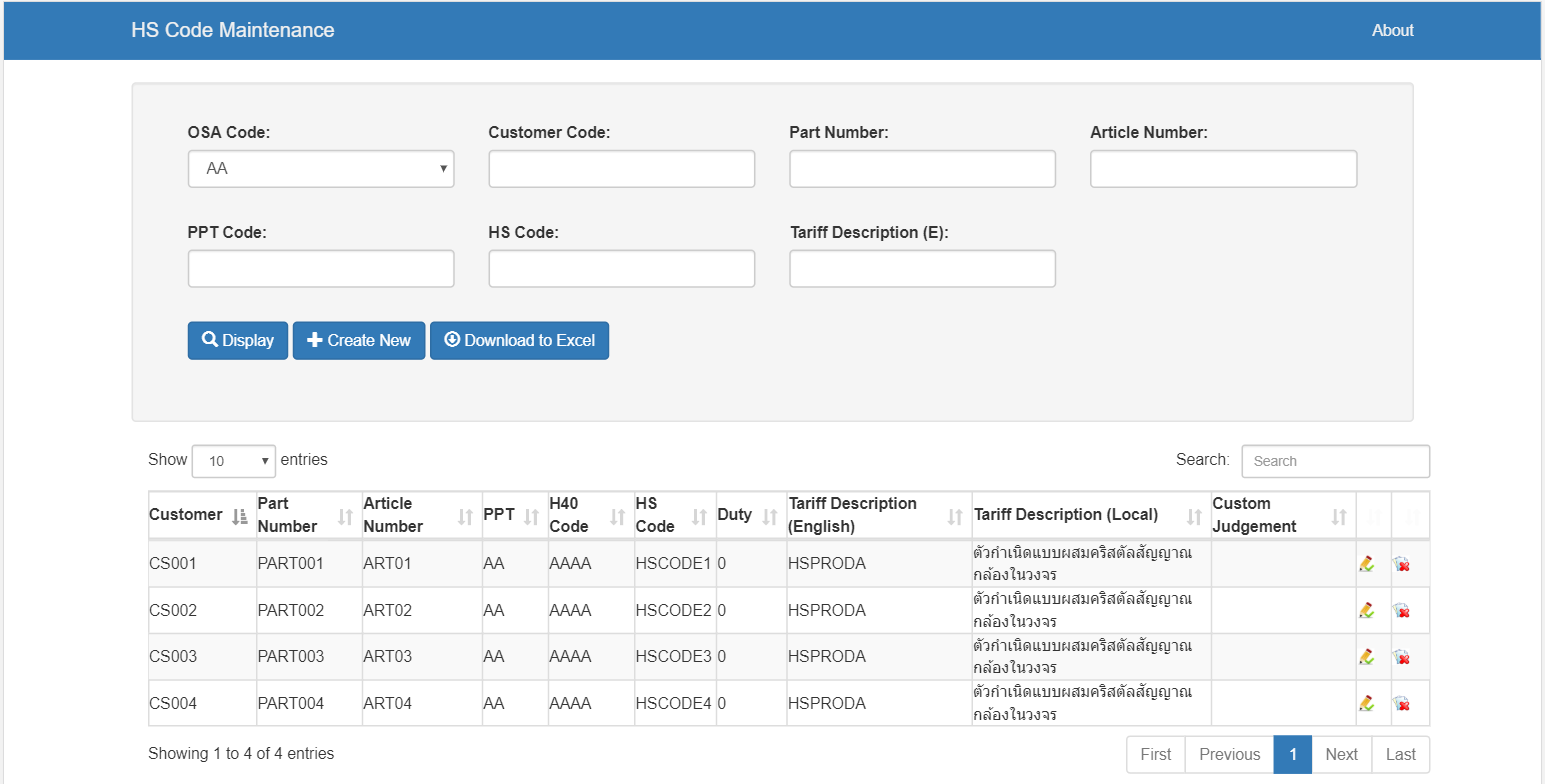
1. Run project ***CRUD\_Example*** in server with configuration as UTF-8 to be able to test the internationalization and input ***main*** in the URL like 🡪 [***http://localhost:8080/CRUD\_Example/***](http://localhost:8080/CRUD_Example/)***main***



This is where the UTF-8 configuration can be found.



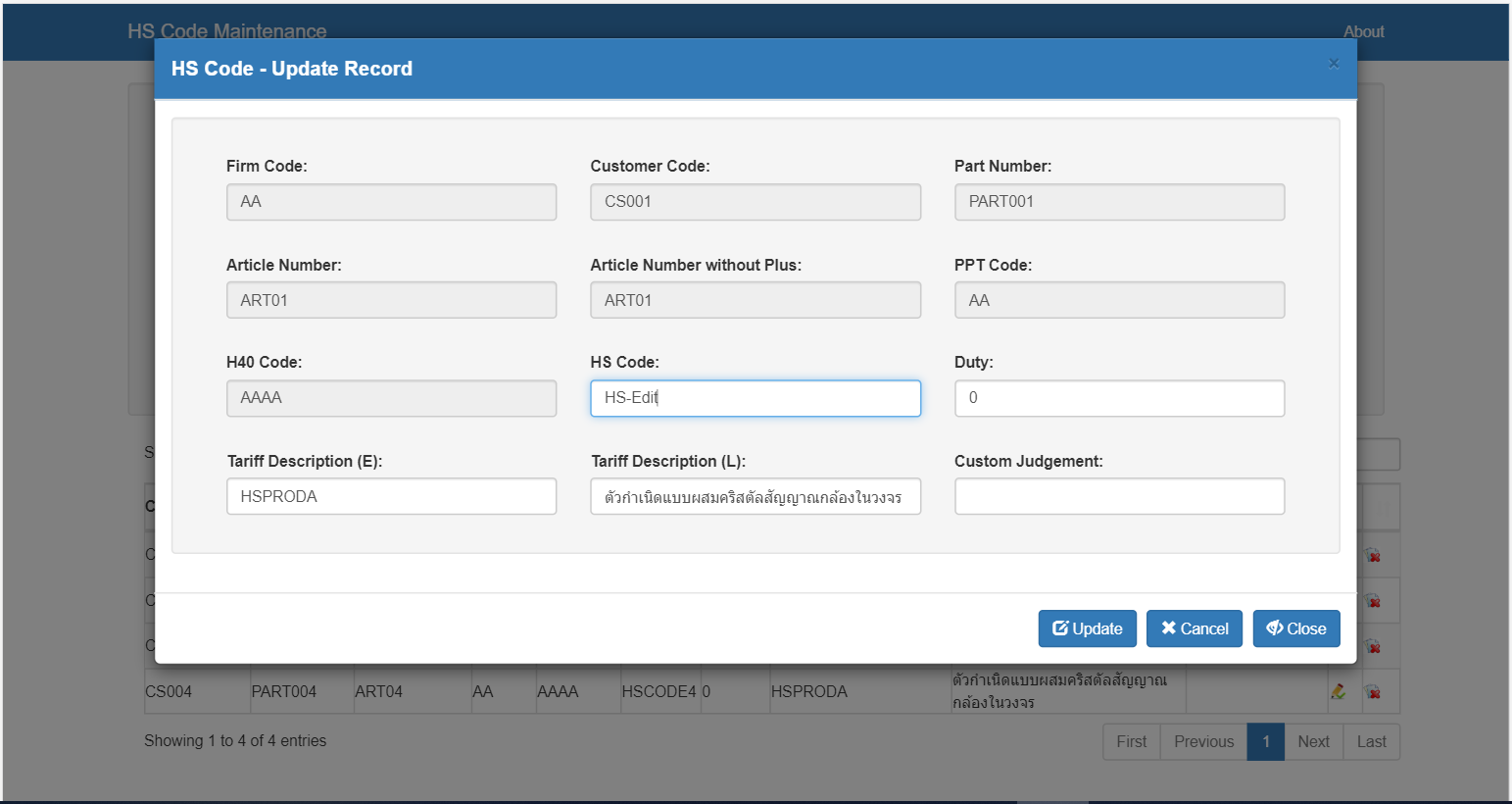
1. Example Screen when *“Display*” button is click.



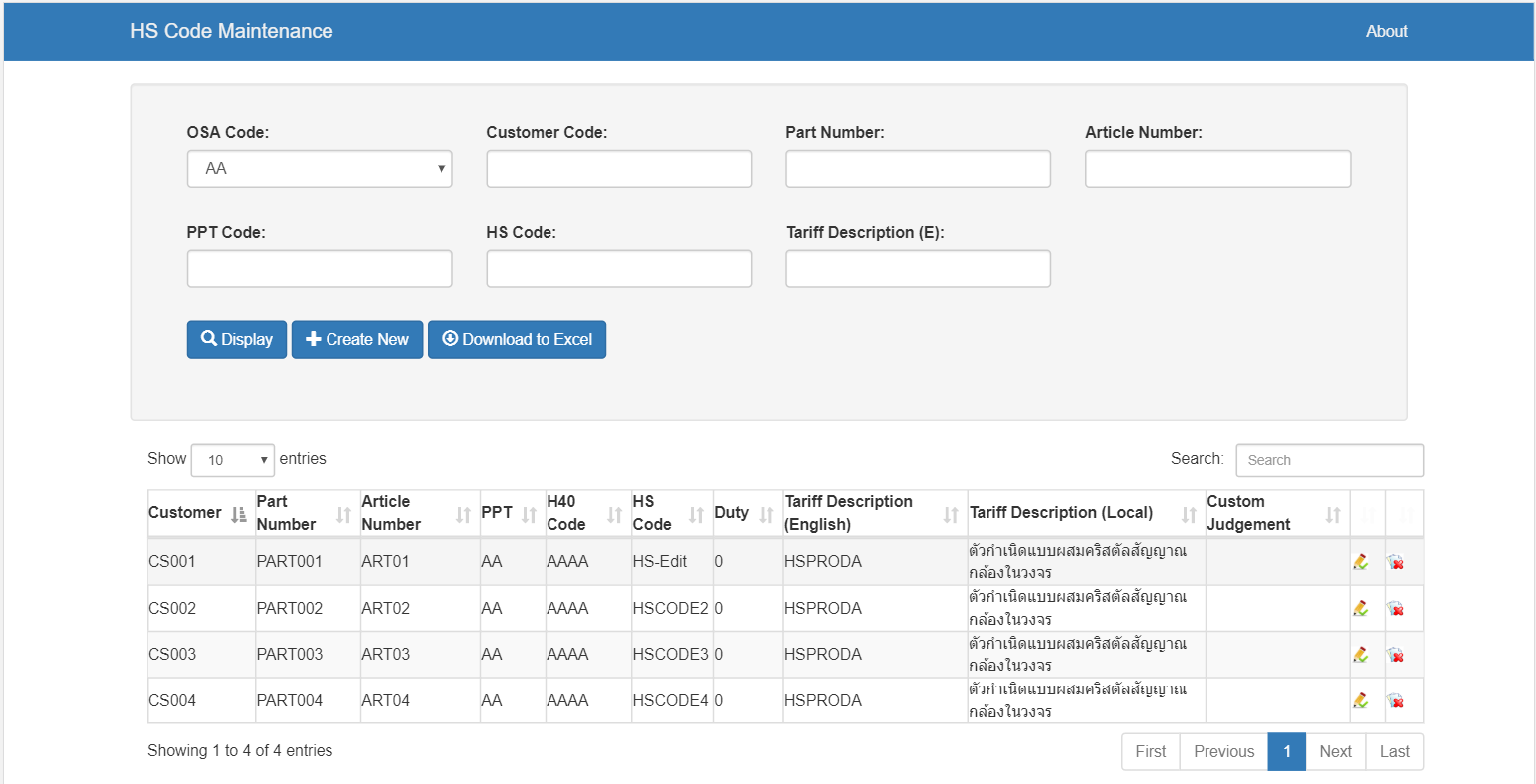
1. After clicking “Display” the above screen will be shown. After the column *“Custom Judgment”* field, there are these buttons:

*  - for editing the record
*  - for data deletion

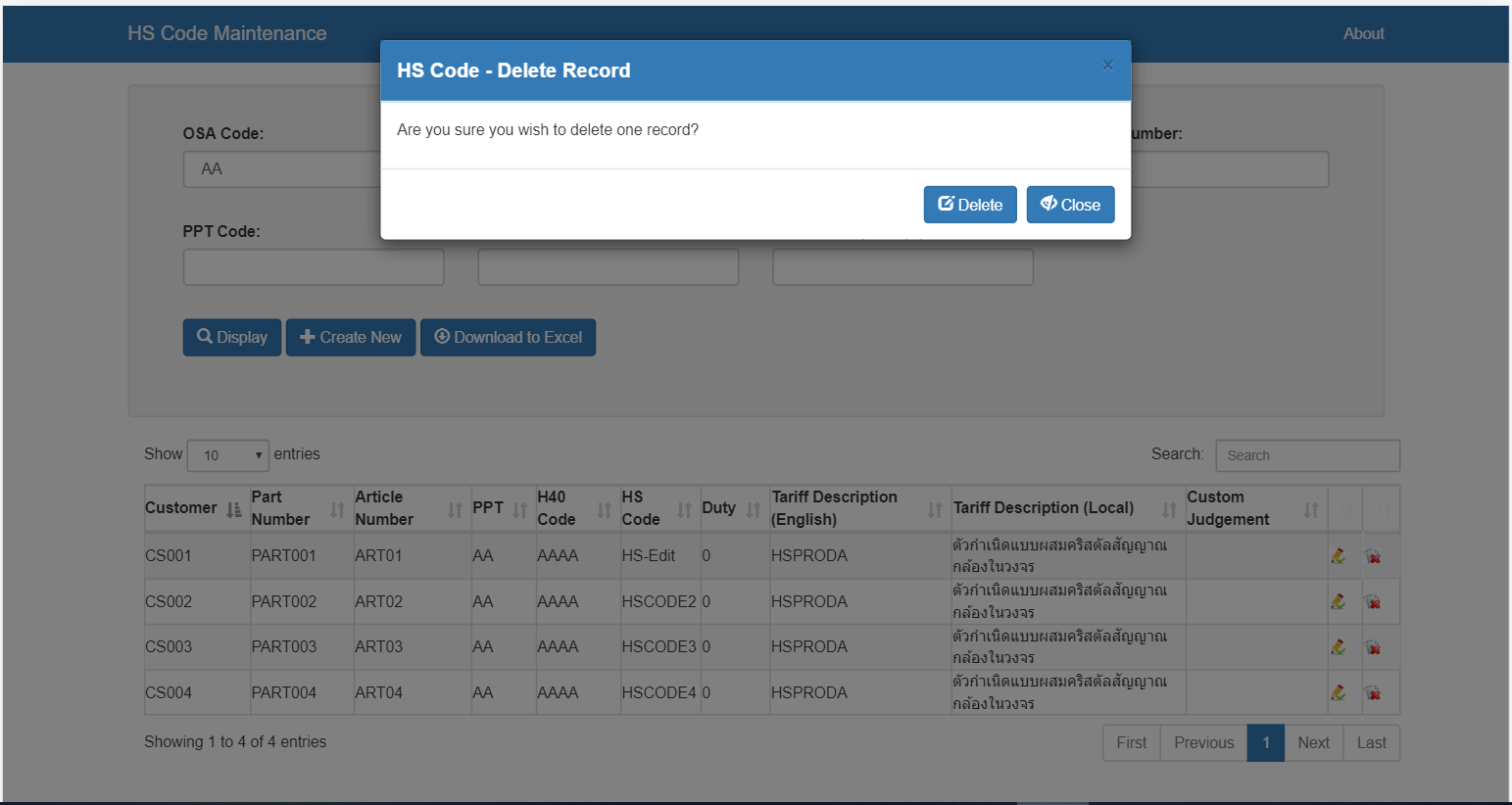
1. This is the screen when edit () button is click. I will edit the HS Code field from “HSCODE1” to “HS-Edit” and click Update.



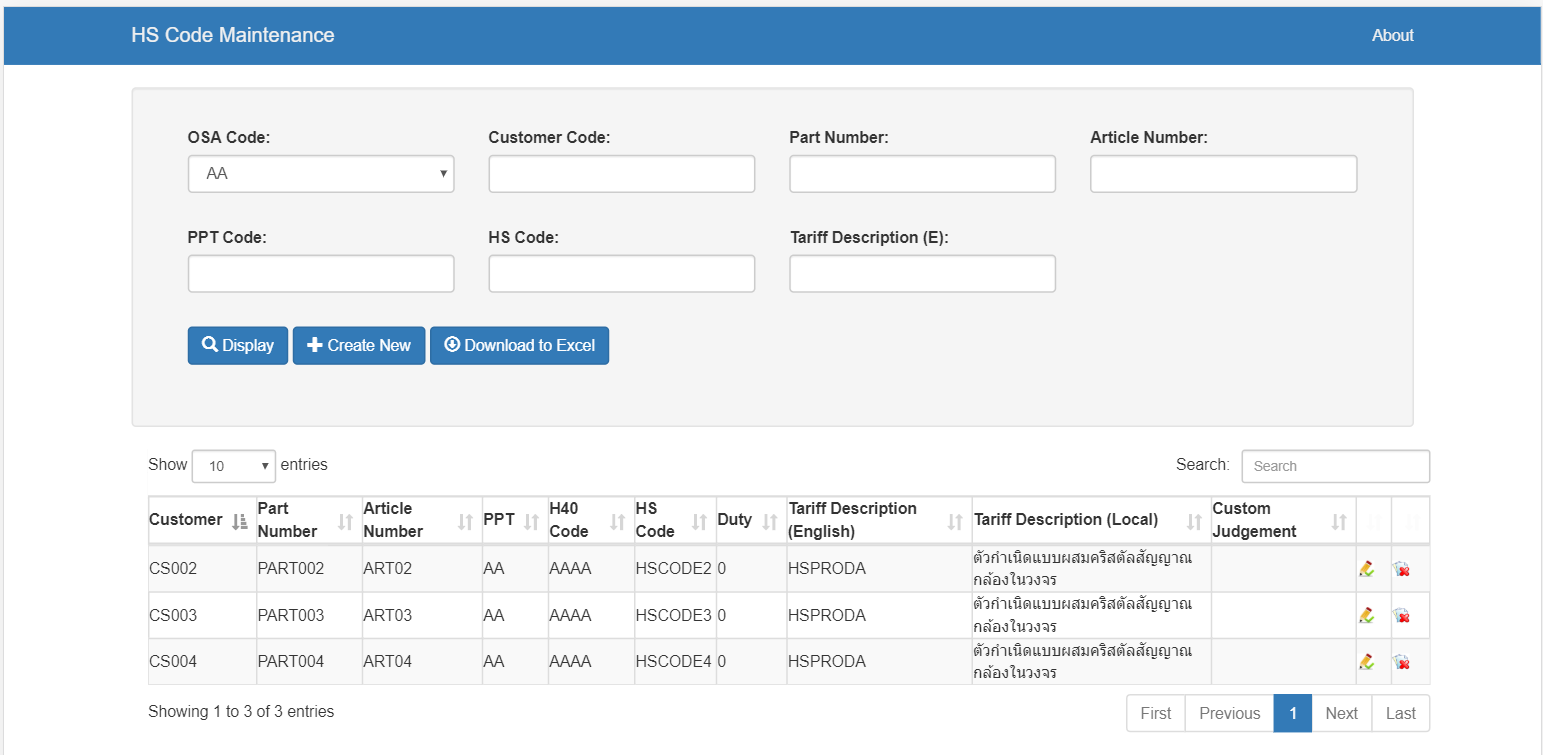
The HS Code is change to HS-Edit as shown below:



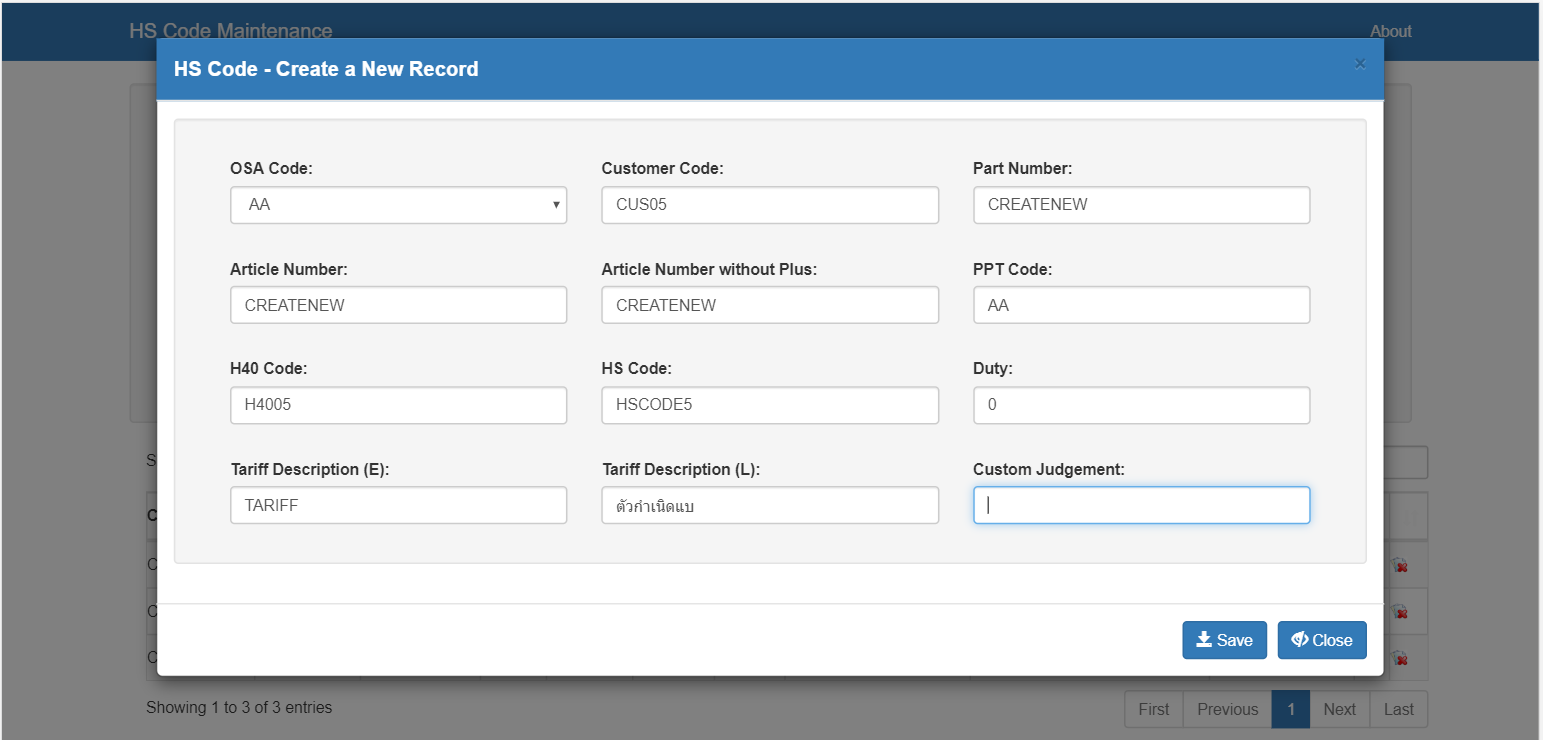
1. This is the screen when delete () button is click. I will delete the same record which I edited above.



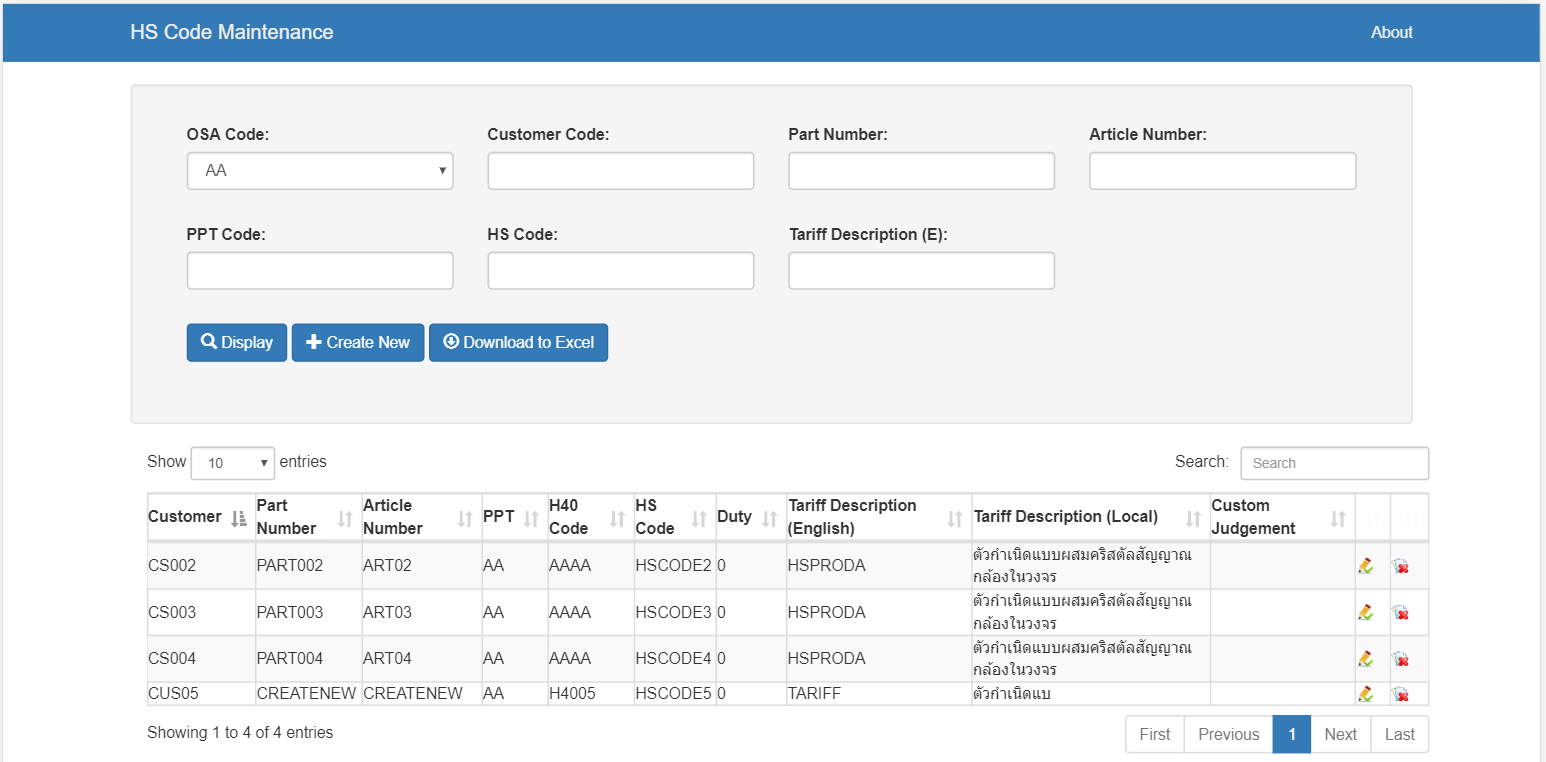
After confirmed that want to delete the record the below screen will be shown, showing the record has been deleted.



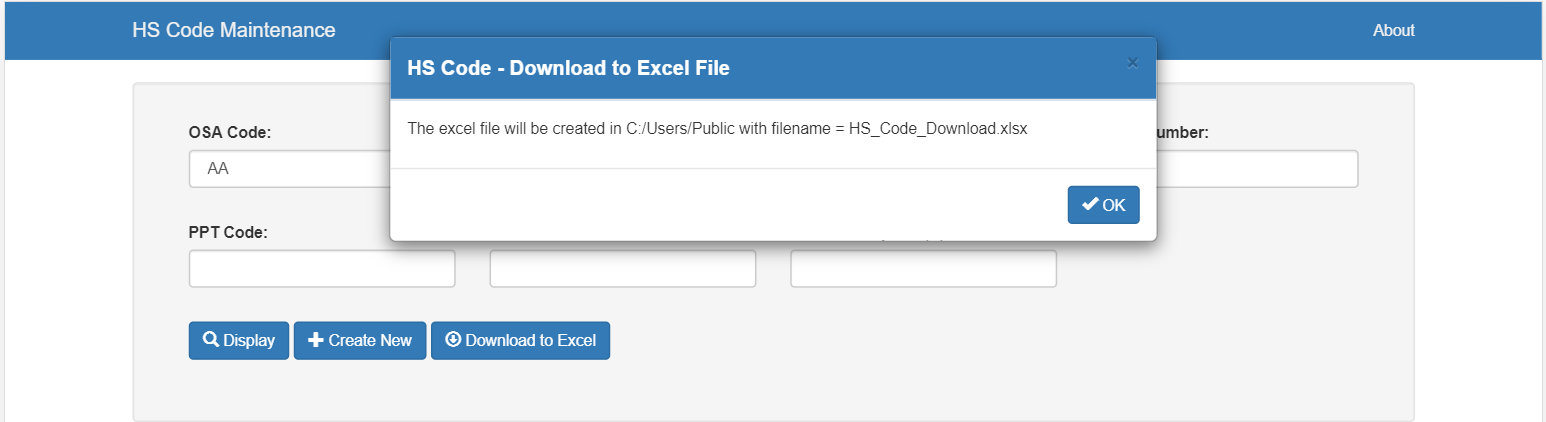
1. Click “*Create New*” button and the below screen will be displayed. I inputted the following values and click “*Save*”.



1. After clicking “*Save*” you can see that the record is created.

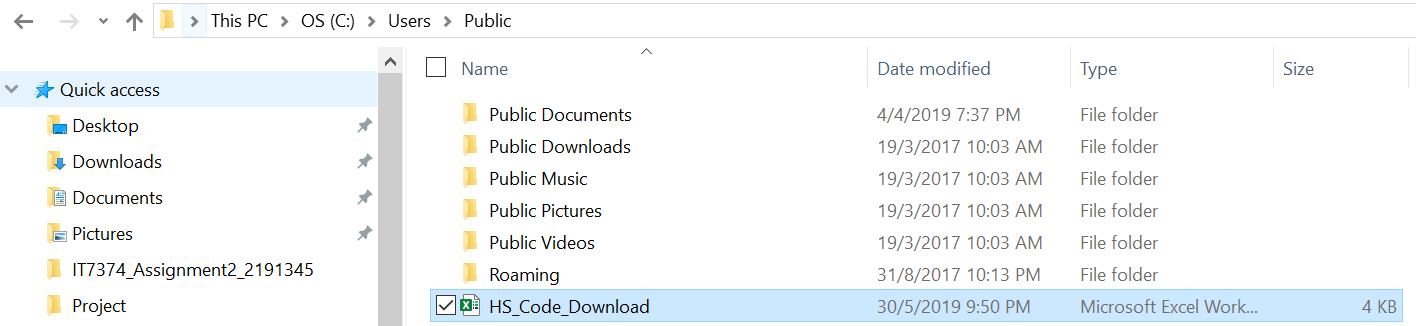


1. Clicking “*Download to Excel*” button will prompt the below screen.

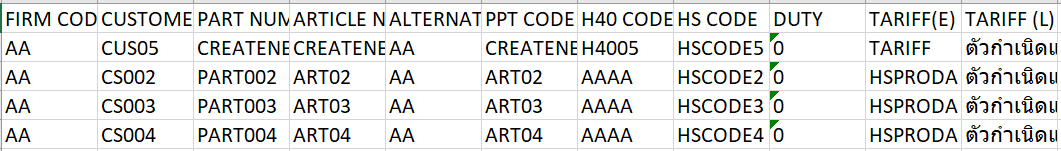


Press “*OK*” to continue downloading and the file will be created in C:/Users/Public with filename = “*HS\_Code\_Download.xlsx*”.

Directory showing the excel file is created.

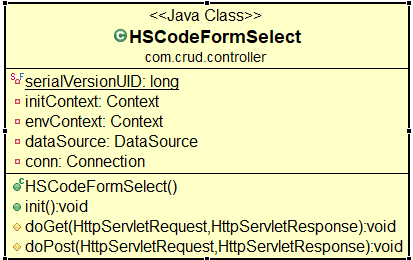


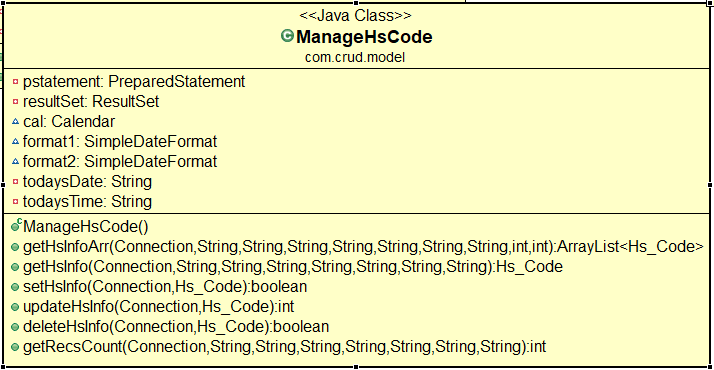
Excel File Created.

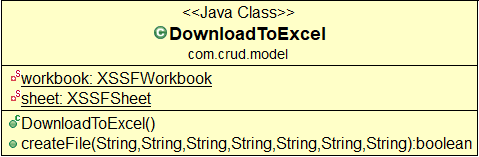


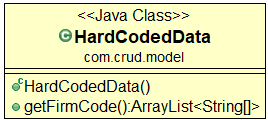
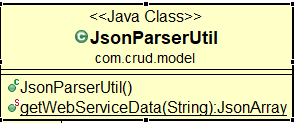
**Class Diagrams:**

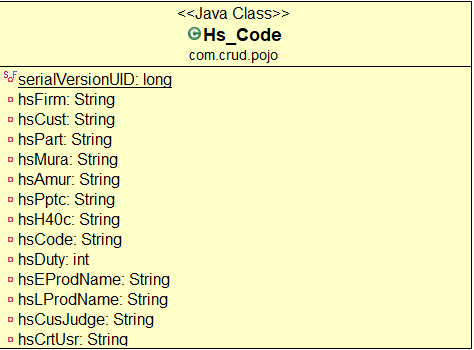
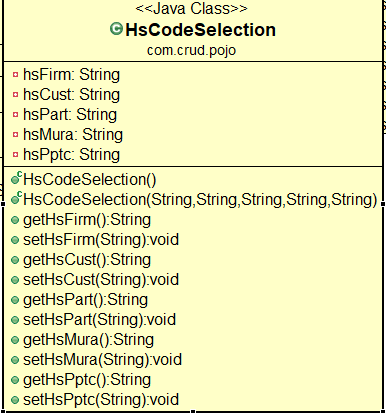
1. **CRUD\_Example**











1. **CRUD\_Example\_Web\_Service**

