INTELLIGENT SYSTEMS

(Part Four – Rule Based Expert System and Recommender System)

Version: 0.1 (20181114T1436)

Author:

Ms Juan Antonio Castro Silva (juan.castro@usco.edu.co)

PhD Diego Hernán Peluffo Ordoñez

1. INTRODUCTION

In this tutorial we are going to build an ecommerce system in java with a PostgreSQL database. We will use Bootstrap for the Frontend and Java Web Services for the Backend. The system will run in the WildFly Application Server.

The ecommerce project will have two intelligent components:

- A coupon generator builds with the PyKnow library.
- A recommender system builds with turicreate framework.

Note: In the GitHub repository you will find the source code, the SQL scripts and the database backup of this project

Basics

An expert system is a program capable of pairing up a set of **facts** with a set of **rules** to those facts, and execute some actions based on the matching rules.

SOFTWARE INSTALLATION

PyKnow: Expert Systems for Python

PyKnow is a Python library for building expert systems strongly inspired by CLIPS.

To install PyKnow, run this command in your terminal:

```
$ pip install pyknow
```

Turi Create

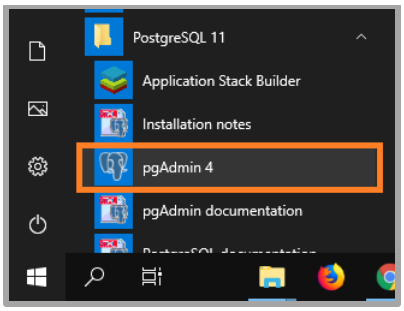
Turi Create simplifies the development of custom machine learning models. You don't have to be a machine learning expert to add recommendations, object detection, image classification, image similarity or activity classification to your app.

To install turicreate, run this command in your terminal:

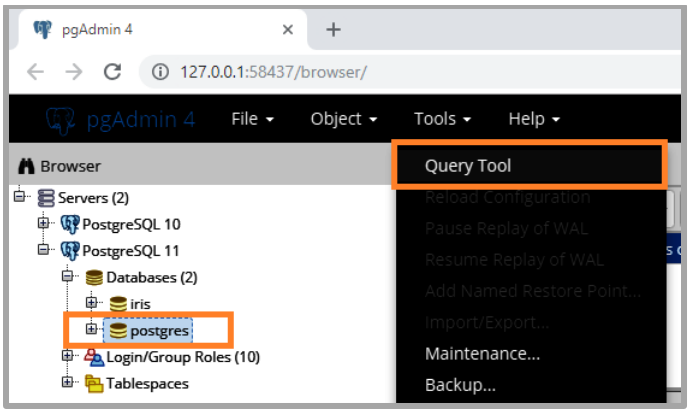
```
$ pip install turicreate
```

DATABASE CREATION

Open pgAdmin4 to create the database.



To execute the SQL queries, click the [Tools Menu] and select the Query Tool.

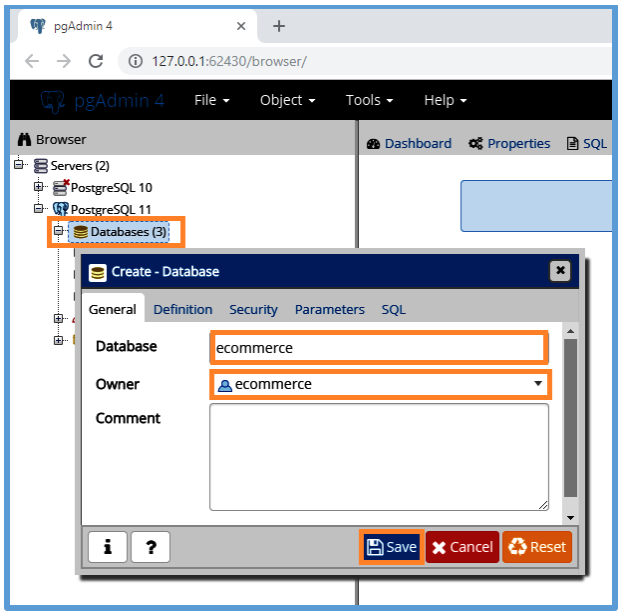


Create a role to be the database owner, execute this command:

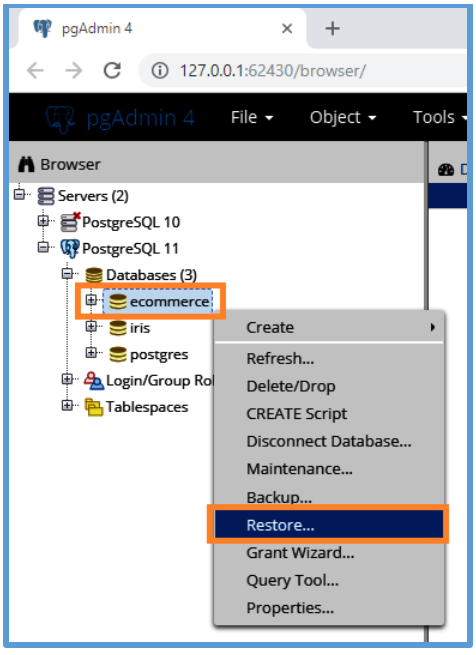
```
CREATE ROLE ecommerce WITH LOGIN ENCRYPTED PASSWORD 'ecommerce';
```

In the Query Tool copy the create role command and click the execute button.

You can create the database objects using the PostgreSQL GUI tools, right click on the Databases folder and select the Create, Database option. In the Create – Database window, write the database name (ecommerce), select the owner (ecommerce) and click the [Save] button.

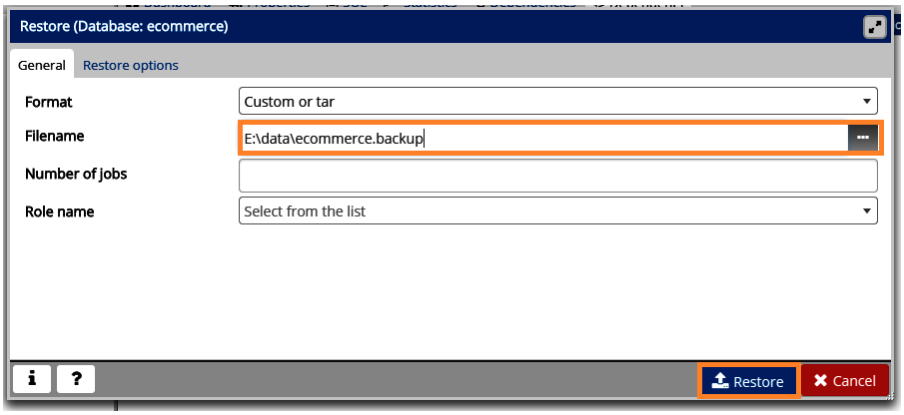


To restore the database backup (ecommerce.backup), right click the ecommerce database and select the [Restore…] option.



In the Restore Database window, click the browse button, select the filename (ecommerce.backup) and click the [Restore] button.

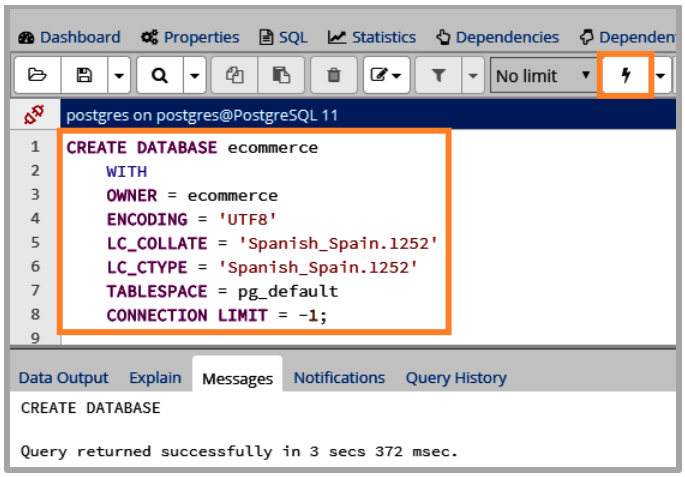
Note: the ecommerce.backup file is in the GitHub repository.



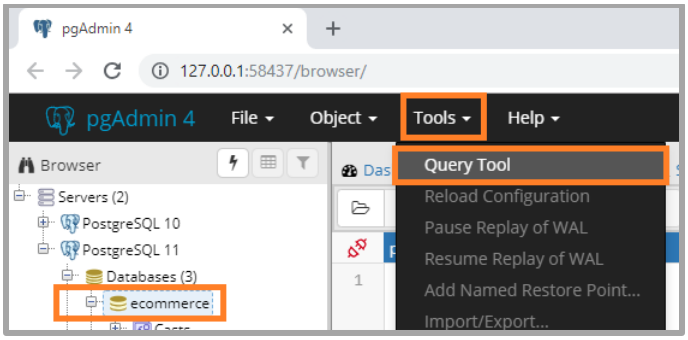
If you want to create the database objects manually, execute the create database commend:

```
CREATE DATABASE ecommerce
    WITH
    OWNER = ecommerce
    ENCODING = 'UTF8'
    LC_COLLATE = 'Spanish_Spain.1252'
    LC_CTYPE = 'Spanish_Spain.1252'
    TABLESPACE = pg_default
    CONNECTION LIMIT = -1;
```

In the Query Tool copy the create database command and click the execute button.



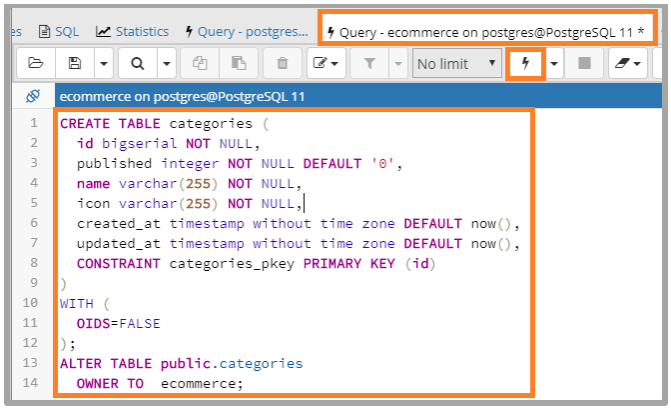
Select the ecommerce database and open the Query Tool.



Create the categories table, execute the create table command:

```
CREATE TABLE categories (
  id bigserial NOT NULL,
  published integer NOT NULL DEFAULT '0',
  name varchar (255) NOT NULL,
  icon varchar (255) NOT NULL,
  created_at timestamp without time zone DEFAULT now (),
  updated_at timestamp without time zone DEFAULT now (),
  CONSTRAINT categories_pkey PRIMARY KEY (id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.categories
  OWNER TO ecommerce;
```

In the Query Tool copy the create table command and click the execute button.



Create the products table.

```
CREATE TABLE products (
  id bigserial NOT NULL,
  published integer NOT NULL DEFAULT '0',
  rating_cache double precision NOT NULL DEFAULT '3.0',
  rating_count integer NOT NULL DEFAULT '0',
  category id bigint NOT NULL,
  name varchar (255) NOT NULL,
  pricing double precision NOT NULL DEFAULT '0.00',
  short_description varchar (255) NOT NULL,
  long_description text NOT NULL,
  icon varchar (255) NOT NULL,
  created_at timestamp without time zone DEFAULT now (),
  updated_at timestamp without time zone DEFAULT now (),
  CONSTRAINT products_pkey PRIMARY KEY (id),
  CONSTRAINT products_category_id_fkey FOREIGN KEY (category_id)
    REFERENCES public.categories (id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.products
  OWNER TO ecommerce;
```

Create the users table.

```
CREATE TABLE users (
  id bigserial NOT NULL,
  user_type integer NOT NULL DEFAULT '0',
  username varchar (128) NOT NULL,
  email varchar (128) NOT NULL,
  password varchar (128) NOT NULL,
  created_at timestamp without time zone DEFAULT now (),
  updated_at timestamp without time zone DEFAULT now (),
  CONSTRAINT users_pkey PRIMARY KEY (id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.users
  OWNER TO ecommerce;
```

Create the reviews table.

```
CREATE TABLE reviews (
  id bigserial NOT NULL,
  product_id bigint NOT NULL,
  user_id bigint NOT NULL,
  rating double precision NOT NULL,
  comment text NOT NULL,
  approved integer NOT NULL DEFAULT '1',
  spam integer NOT NULL DEFAULT '0',
  created_at timestamp without time zone DEFAULT now (),
  updated_at timestamp without time zone DEFAULT now (),
  CONSTRAINT reviews_pkey PRIMARY KEY (id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE public.reviews
  OWNER TO ecommerce;
```

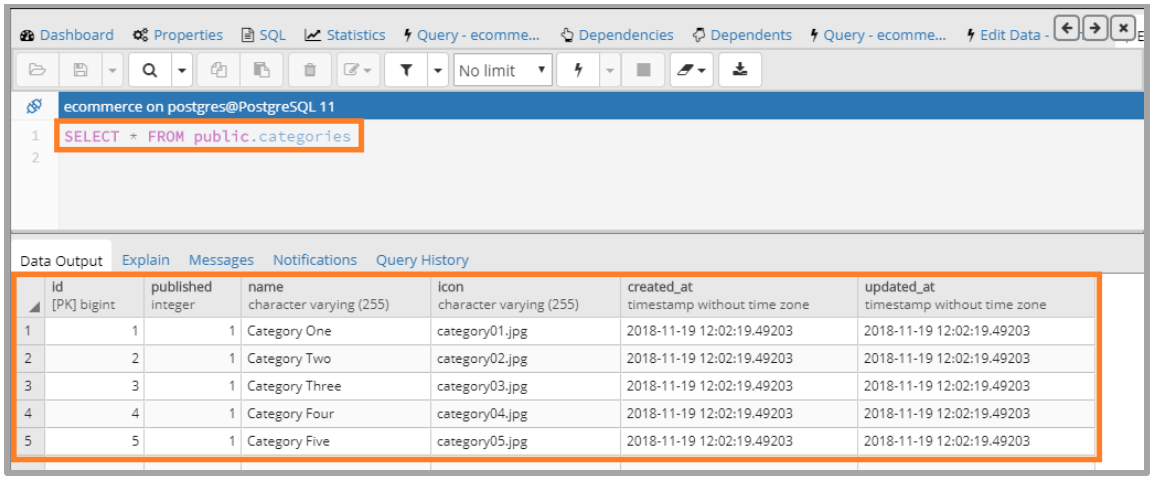
INSERT THE DATA

Open the Query Tool and execute the insert commands.

Categories

```
INSERT INTO categories ("published", "name", "icon") VALUES
(1, 'Category One', 'category01.jpg'),
(1, 'Category Two', 'category02.jpg'),
(1, 'Category Three', 'category03.jpg'),
(1, 'Category Four', 'category04.jpg'),
(1, 'Category Five', 'category05.jpg');
```

Test categories data inserted in the table.



	id [PK] bigint	published integer	name character varying (255)	icon character varying (255)	created_at timestamp without time zone	updated_at timestamp without time zone
1	1	1	Category One	category01.jpg	2018-11-19 12:02:19.49203	2018-11-19 12:02:19.49203
2	2	1	Category Two	category02.jpg	2018-11-19 12:02:19.49203	2018-11-19 12:02:19.49203
3	3	1	Category Three	category03.jpg	2018-11-19 12:02:19.49203	2018-11-19 12:02:19.49203
4	4	1	Category Four	category04.jpg	2018-11-19 12:02:19.49203	2018-11-19 12:02:19.49203
5	5	1	Category Five	category05.jpg	2018-11-19 12:02:19.49203	2018-11-19 12:02:19.49203

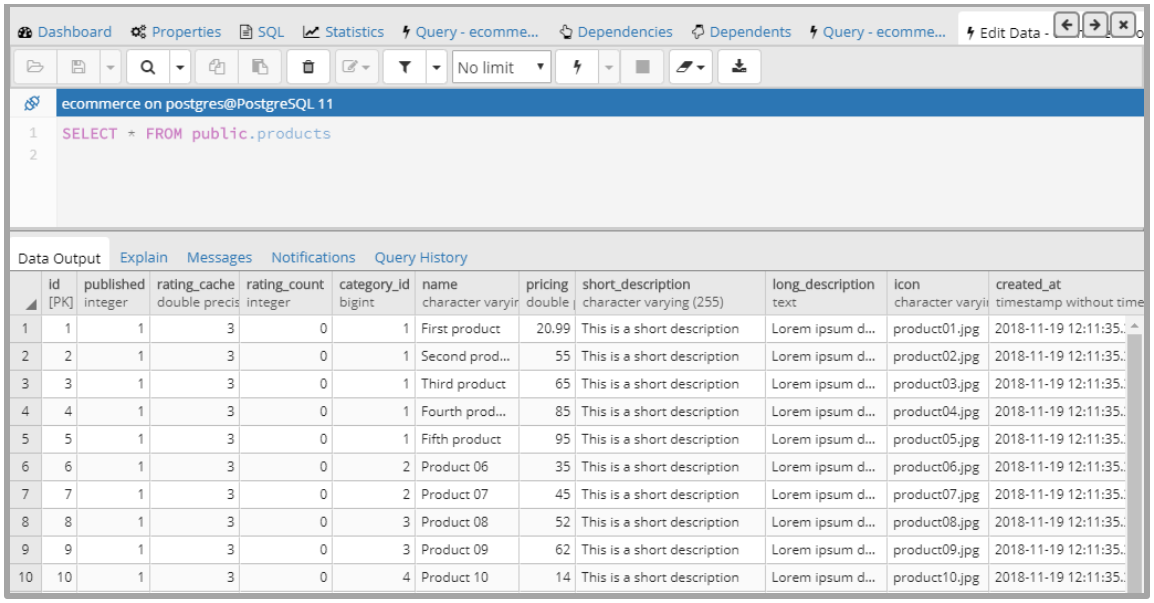
Products

```
INSERT INTO products ("published", "rating_cache", "rating_count", "category_id",
"name", "pricing", "short_description", "long_description", "icon") VALUES

(1, 3.0, 0, 1, 'First product', 20.99, 'This is a short description', 'Lorem ipsum
dolor ...', 'product01.jpg'),
(1, 3.0, 0, 1, 'Second product', 55.00, 'This is a short description', 'Lorem ipsum
dolor ...', 'product02.jpg'),
(1, 3.0, 0, 1, 'Third product', 65.00, 'This is a short description', 'Lorem ipsum
dolor ...', 'product03.jpg'),
(1, 3.0, 0, 1, 'Fourth product', 85.00, 'This is a short description', 'Lorem ipsum
dolor ...', 'product04.jpg'),
(1, 3.0, 0, 1, 'Fifth product', 95.00, 'This is a short description', 'Lorem ipsum
dolor ...', 'product05.jpg'),
(1, 3.0, 0, 2, 'Product 06', 35.00, 'This is a short description', 'Lorem ipsum
dolor ...', 'product06.jpg'),
(1, 3.0, 0, 2, 'Product 07', 45.00, 'This is a short description', 'Lorem ipsum
dolor ...', 'product07.jpg'),
(1, 3.0, 0, 3, 'Product 08', 52.00, 'This is a short description', 'Lorem ipsum
dolor ...', 'product08.jpg'),
(1, 3.0, 0, 3, 'Product 09', 62.00, 'This is a short description', 'Lorem ipsum
dolor ...', 'product09.jpg'),
(1, 3.0, 0, 4, 'Product 10', 14.00, 'This is a short description', 'Lorem ipsum
dolor ...', 'product10.jpg'),
(1, 3.0, 0, 4, 'Product 11', 18.00, 'This is a short description', 'Lorem ipsum
dolor ...', 'product11.jpg'),
(1, 3.0, 0, 5, 'Product 12', 40.00, 'This is a short description', 'Lorem ipsum
dolor ...', 'product12.jpg'),
(1, 3.0, 0, 5, 'Product 13', 44.00, 'This is a short description', 'Lorem ipsum
dolor ...', 'product13.jpg');
```

Update the products long_description field.

```
UPDATE products SET long_description = 'Lorem ipsum dolor sit amet, consectetur
adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex
ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse
cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia deserunt mollit anim id est laborum';
```

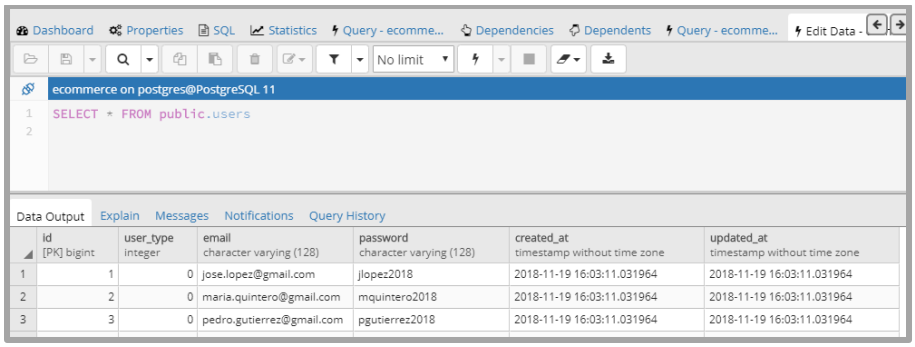


	id [PK] integer	published integer	rating_cache double precis	rating_count integer	category_id bigint	name character varyir	pricing double	short_description character varying (255)	long_description text	icon character varyi	created_at timestamp without time
1	1	1	3	0	1	First product	20.99	This is a short description	Lorem ipsum d...	product01.jpg	2018-11-19 12:11:35.
2	2	1	3	0	1	Second prod...	55	This is a short description	Lorem ipsum d...	product02.jpg	2018-11-19 12:11:35.
3	3	1	3	0	1	Third product	65	This is a short description	Lorem ipsum d...	product03.jpg	2018-11-19 12:11:35.
4	4	1	3	0	1	Fourth prod...	85	This is a short description	Lorem ipsum d...	product04.jpg	2018-11-19 12:11:35.
5	5	1	3	0	1	Fifth product	95	This is a short description	Lorem ipsum d...	product05.jpg	2018-11-19 12:11:35.
6	6	1	3	0	2	Product 06	35	This is a short description	Lorem ipsum d...	product06.jpg	2018-11-19 12:11:35.
7	7	1	3	0	2	Product 07	45	This is a short description	Lorem ipsum d...	product07.jpg	2018-11-19 12:11:35.
8	8	1	3	0	3	Product 08	52	This is a short description	Lorem ipsum d...	product08.jpg	2018-11-19 12:11:35.
9	9	1	3	0	3	Product 09	62	This is a short description	Lorem ipsum d...	product09.jpg	2018-11-19 12:11:35.
10	10	1	3	0	4	Product 10	14	This is a short description	Lorem ipsum d...	product10.jpg	2018-11-19 12:11:35.

Users

```
INSERT INTO users ("username", "email", "password") VALUES
('jose', 'jose.lopez@gmail.com', 'jlopez2018'),
('maria', 'maria.quintero@gmail.com', 'mquintero2018'),
('pedro', 'pedro.gutierrez@gmail.com', 'pgutierrez2018');
```

Test the users inserted records.

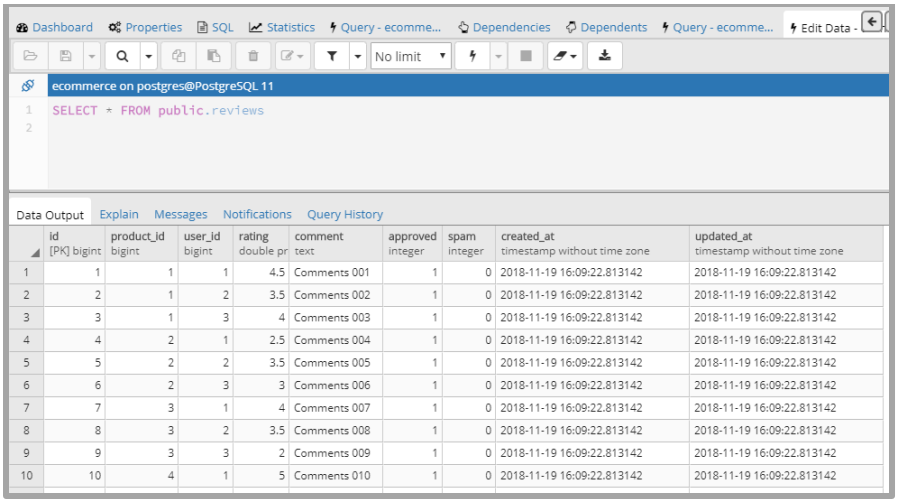


	id [PK] bigint	user_type integer	email character varying (128)	password character varying (128)	created_at timestamp without time zone	updated_at timestamp without time zone
1	1	0	jose.lopez@gmail.com	jlopez2018	2018-11-19 16:03:11.031964	2018-11-19 16:03:11.031964
2	2	0	maria.quintero@gmail.com	mquintero2018	2018-11-19 16:03:11.031964	2018-11-19 16:03:11.031964
3	3	0	pedro.gutierrez@gmail.com	pgutierrez2018	2018-11-19 16:03:11.031964	2018-11-19 16:03:11.031964

Reviews

```
INSERT INTO reviews ("product_id", "user_id", "rating", "comment") VALUES
(1, 1, 4.5, 'Comments 001'),
(1, 2, 3.5, 'Comments 002'),
(1, 3, 4.0, 'Comments 003'),
(2, 1, 2.5, 'Comments 004'),
(2, 2, 3.5, 'Comments 005'),
(2, 3, 3.0, 'Comments 006'),
(3, 1, 4.0, 'Comments 007'),
(3, 2, 3.5, 'Comments 008'),
(3, 3, 2.0, 'Comments 009'),
(4, 1, 5.0, 'Comments 010'),
(4, 2, 3.0, 'Comments 011'),
(4, 3, 3.5, 'Comments 012'),
(5, 1, 3.0, 'Comments 013'),
(5, 2, 4.5, 'Comments 014'),
(5, 3, 4.0, 'Comments 015');
```

Test the reviews inserted records.



	id [PK] bigint	product_id bigint	user_id bigint	rating double pr	comment text	approved integer	spam integer	created_at timestamp without time zone	updated_at timestamp without time zone
1	1	1	1	4.5	Comments 001	1	0	2018-11-19 16:09:22.813142	2018-11-19 16:09:22.813142
2	2	1	2	3.5	Comments 002	1	0	2018-11-19 16:09:22.813142	2018-11-19 16:09:22.813142
3	3	1	3	4	Comments 003	1	0	2018-11-19 16:09:22.813142	2018-11-19 16:09:22.813142
4	4	2	1	2.5	Comments 004	1	0	2018-11-19 16:09:22.813142	2018-11-19 16:09:22.813142
5	5	2	2	3.5	Comments 005	1	0	2018-11-19 16:09:22.813142	2018-11-19 16:09:22.813142
6	6	2	3	3	Comments 006	1	0	2018-11-19 16:09:22.813142	2018-11-19 16:09:22.813142
7	7	3	1	4	Comments 007	1	0	2018-11-19 16:09:22.813142	2018-11-19 16:09:22.813142
8	8	3	2	3.5	Comments 008	1	0	2018-11-19 16:09:22.813142	2018-11-19 16:09:22.813142
9	9	3	3	2	Comments 009	1	0	2018-11-19 16:09:22.813142	2018-11-19 16:09:22.813142
10	10	4	1	5	Comments 010	1	0	2018-11-19 16:09:22.813142	2018-11-19 16:09:22.813142

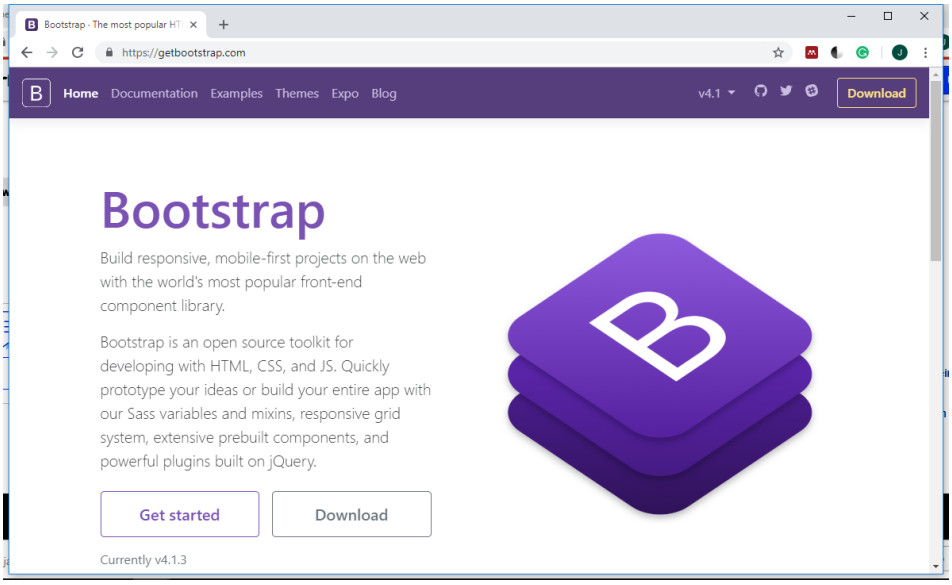
FRONT END – SOFTWARE

To build the Frontend we use Bootstrap and jQuery and some free templates.

Note: All these files are included in the GitHub repository; you do not need to download these files.

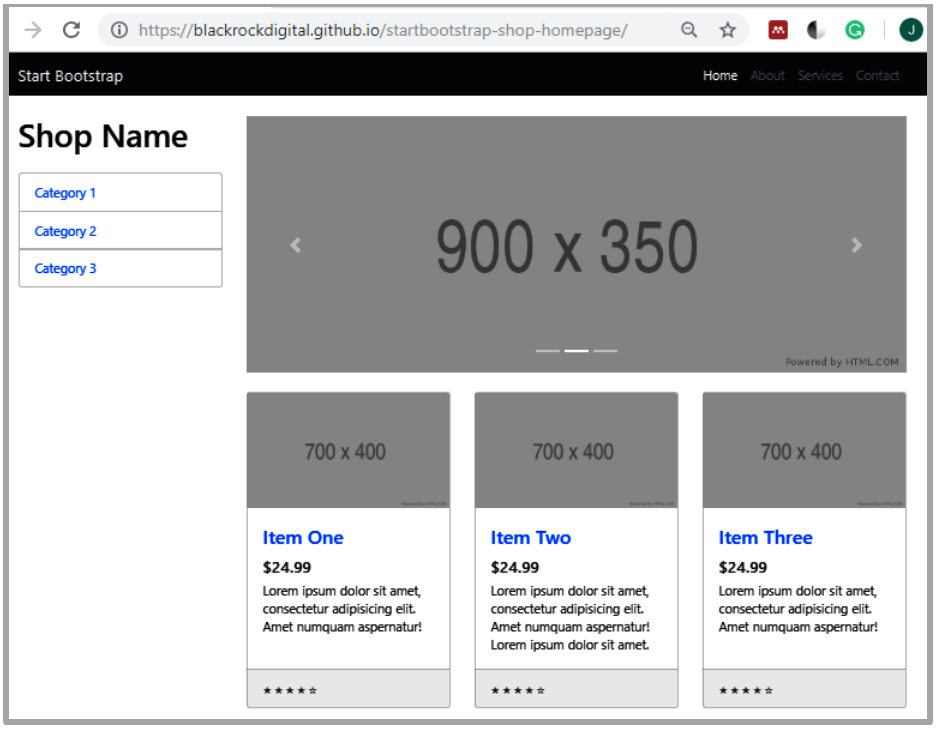
BOOTSTRAP

Bootstrap is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.



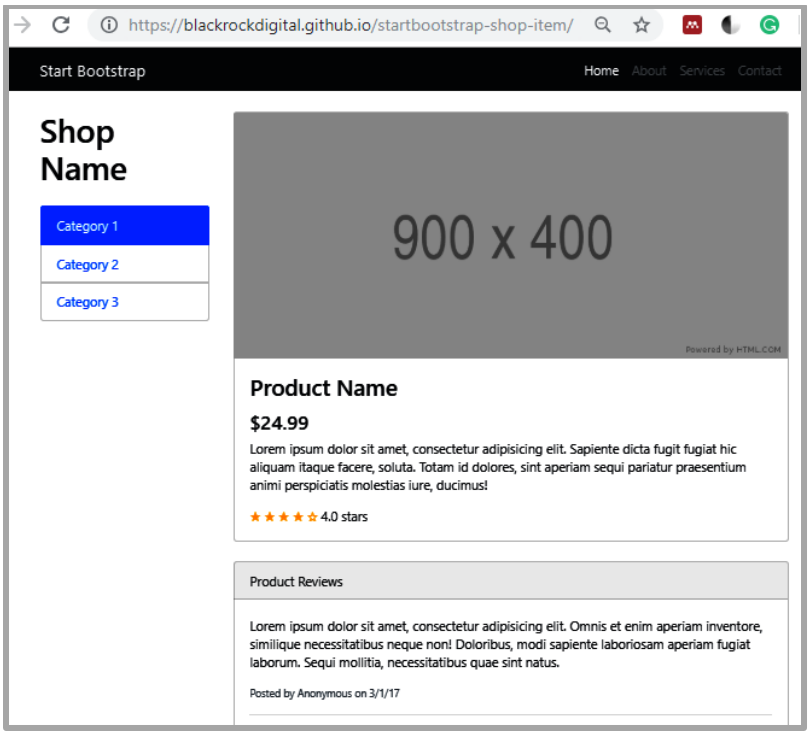
https://getbootstrap.com/

SHOP HOME PAGE



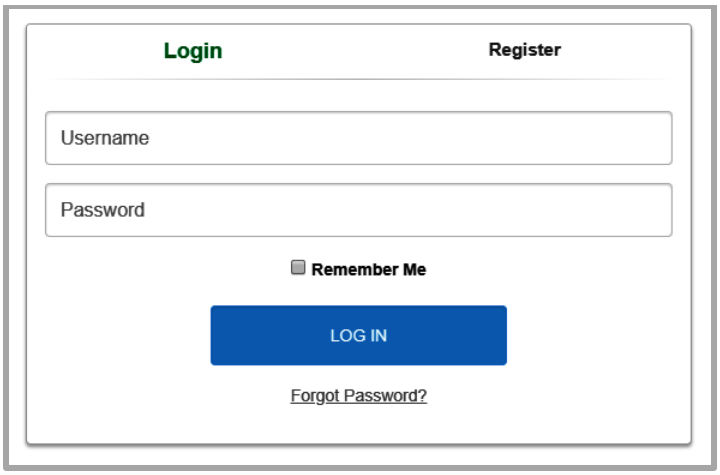
https://startbootstrap.com/template-overviews/shop-homepage/

SHOP ITEM PAGE



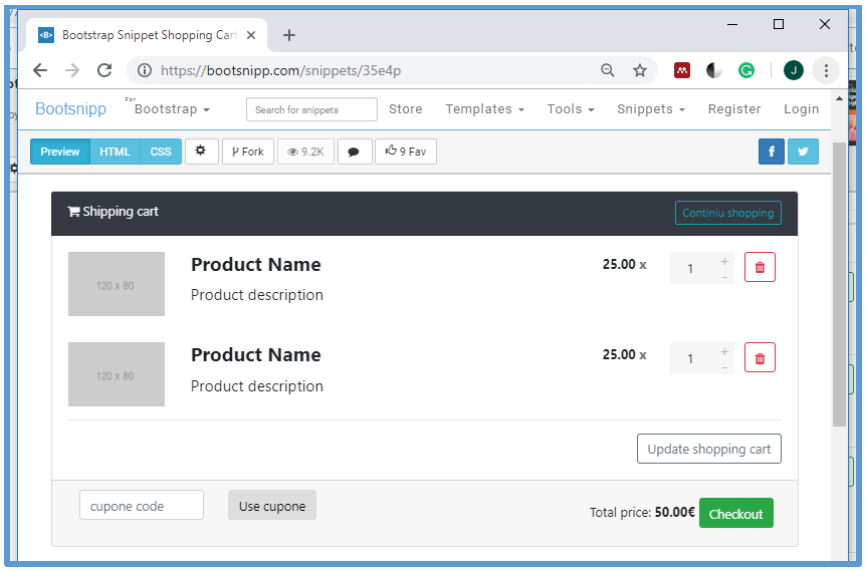
https://startbootstrap.com/template-overviews/shop-item/

LOGIN – REGISTER FORM



https://bootsnipp.com/snippets/featured/login-and-register-tabbed-form

SHOPPING CART

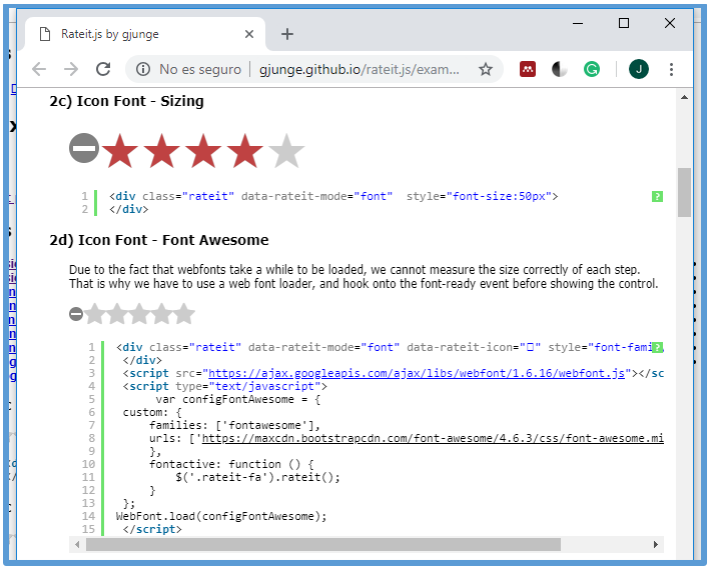


https://bootsnipp.com/snippets/35e4p

RATEIT – START RATING

Rating plugin for jQuery. Fast, Progressive enhancement, touch support, icon-font support, highly customizable, unobtrusive JavaScript (using HTML5 data-* attributes), RTL support, supports as many stars as you'd like, and also any step size.

https://github.com/gjunge/rateit.js



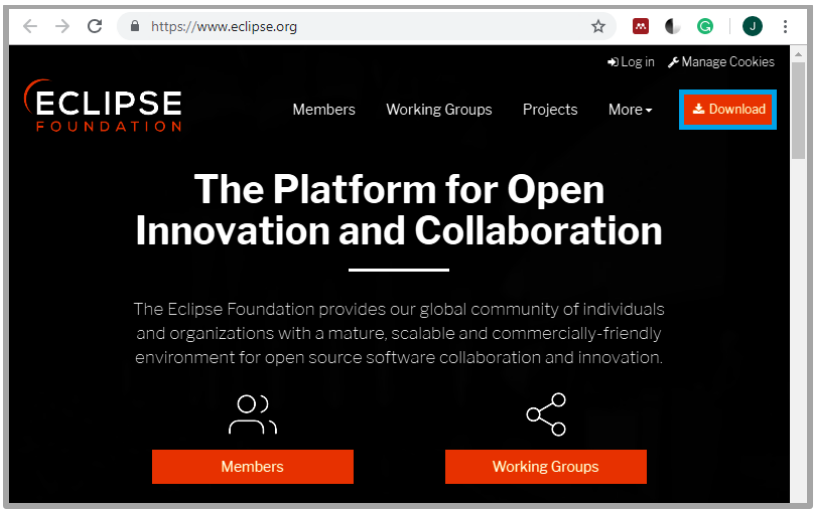
http://gjunge.github.io/rateit.js/examples/

JAVA WEB APPLICACTION PROJECT

In this tutorial we are going to download the eclipse project (ecommerce) from the GitHub repository and import it in the eclipse IDE.

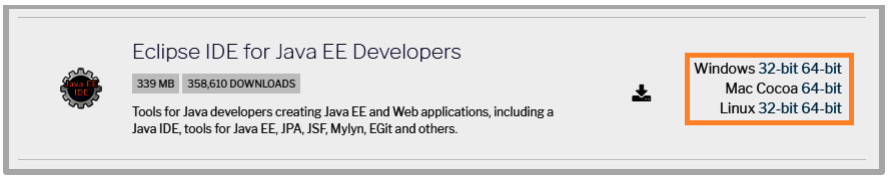
ECLIPSE IDE

To download the Eclipse IDE, open the official web page (http://www.eclipse.org) and click the Download button.

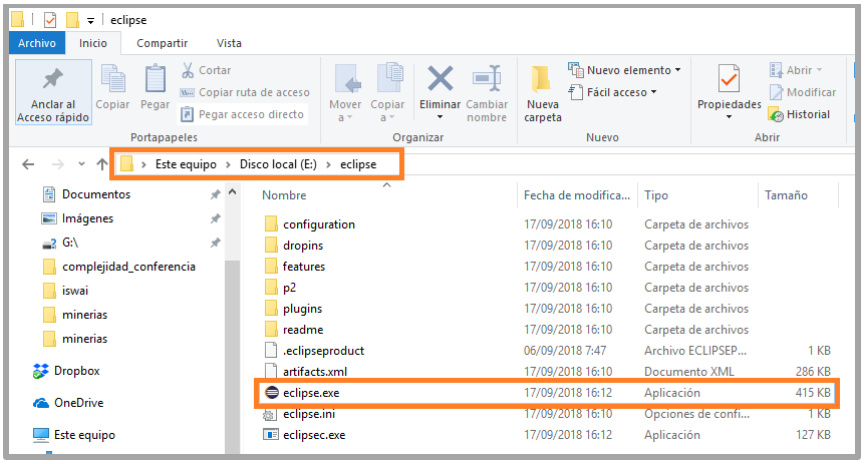


Click the Download Packages link.

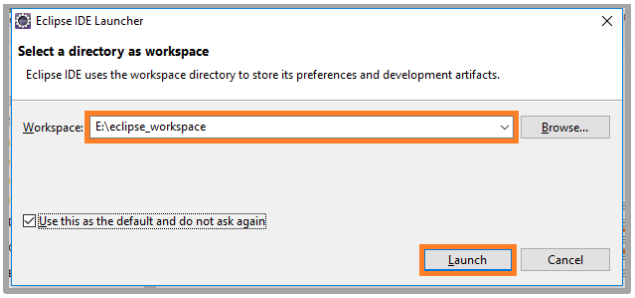
Select the Eclipse IDE for Java EE Developers and click the link corresponding to your platform.



For Windows, unzip the eclipse software and click the eclipse.exe file to start the eclipse IDE.

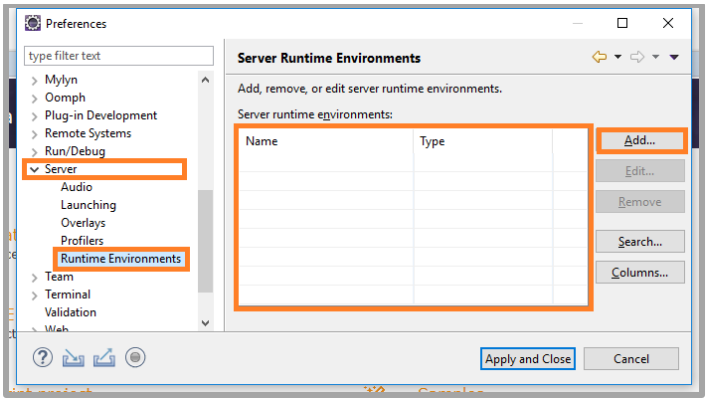


Create or select a folder as workspace.

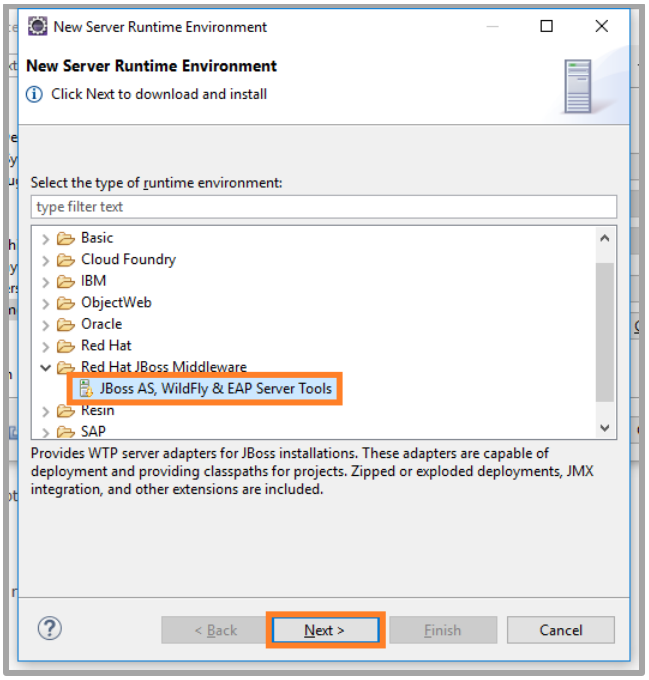


SERVER RUNTIME

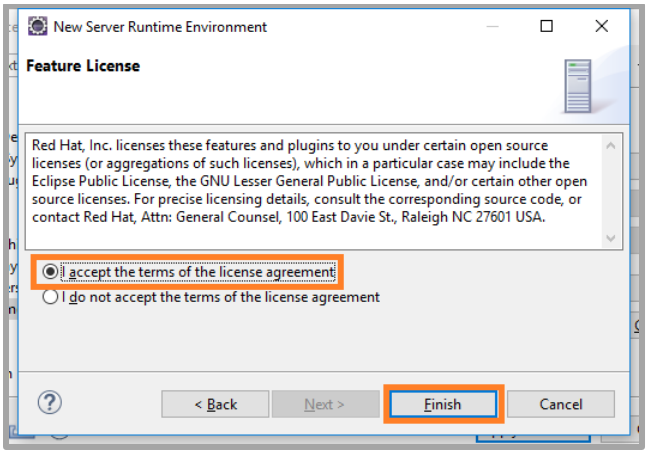
To add a Server Runtime Environment, select the Window menu and click the Preferences option. Select the Server Folder and click the Runtime Environment. Click the [Add…] button.



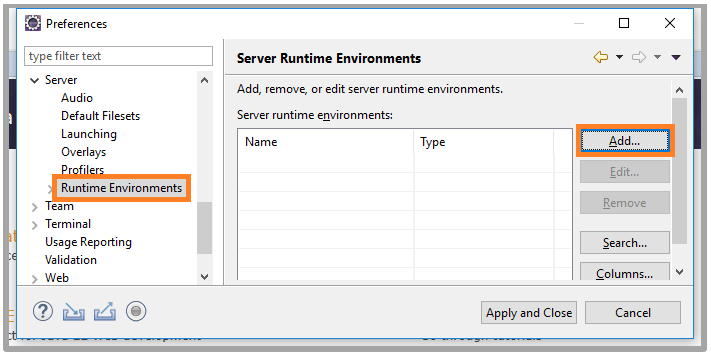
In the Red Hat Jboss Middleware folder, select the Jboss AS, WildFly & EAP Server Tools and click the [Next] button.



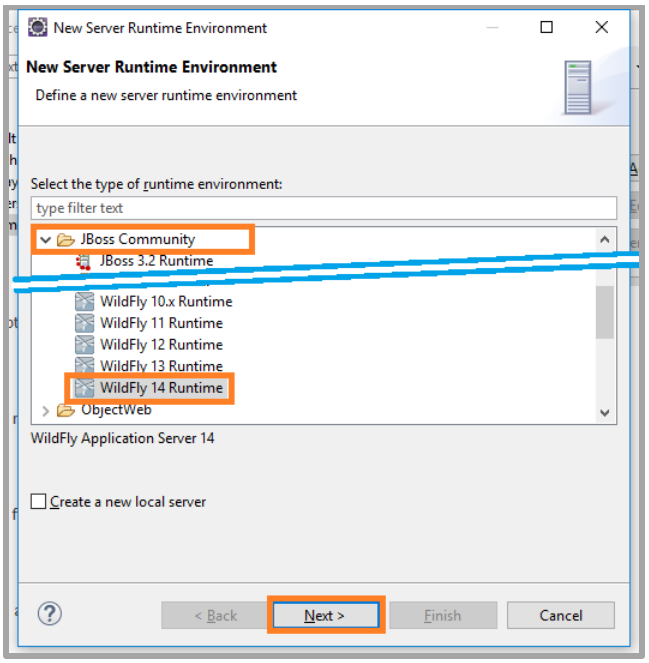
Select the accept the terms of the license agreement option and click the [Finish] button.



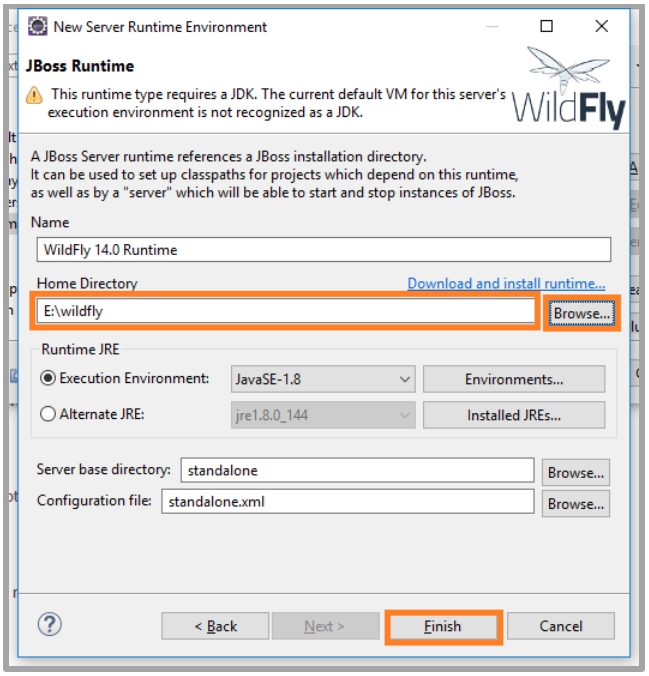
After the installation process, click the [Add…] button.



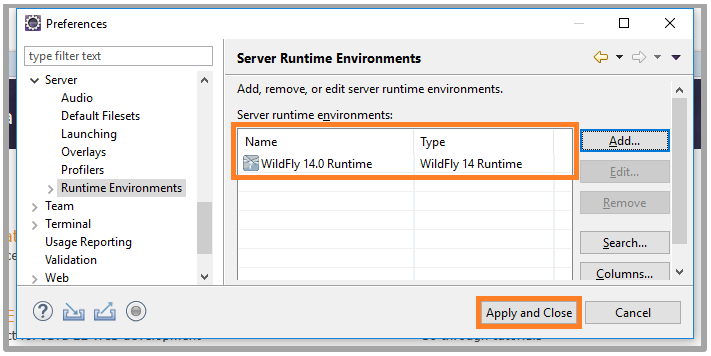
In the Jboss Community, select the WildFly 14 Runtime and click the [Next] button.



Select the Home Directory, the folder where the WildFly is installed and click the [Finish] button.

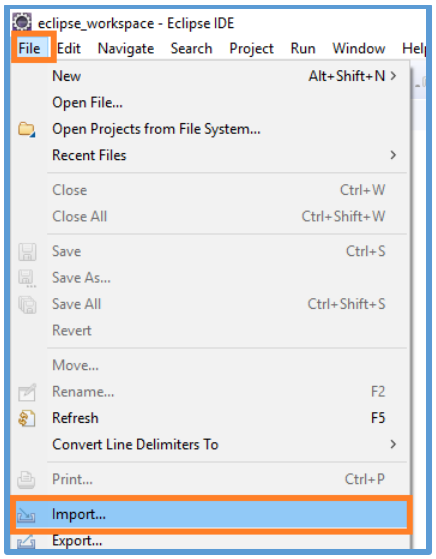


In the Server Runtime Environment you will see the WildFly 14 Runtime, click the [Apply and Close] button.

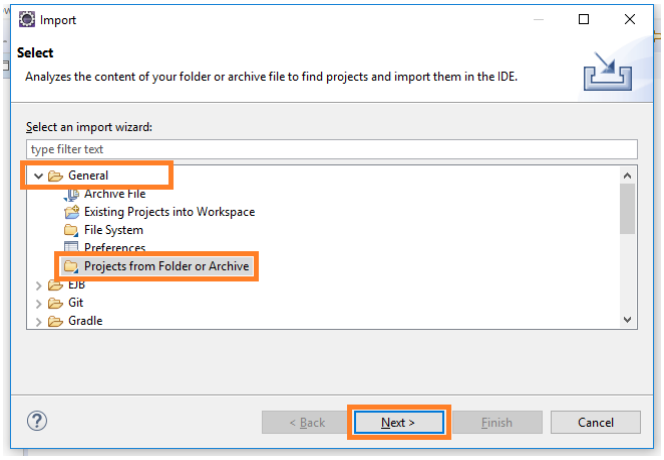


IMPORT THE ECOMMERCE ECLIPSE PROJECT

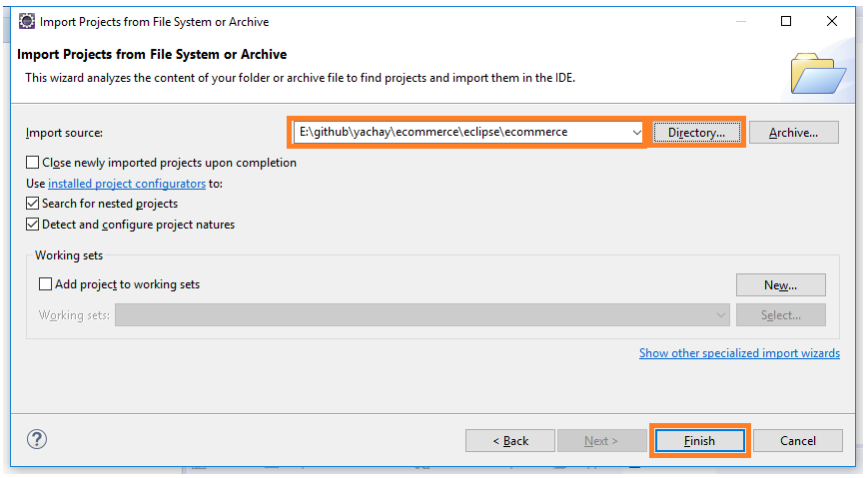
To import the ecommerce project, select the [File] menu and click the [Import …] option.



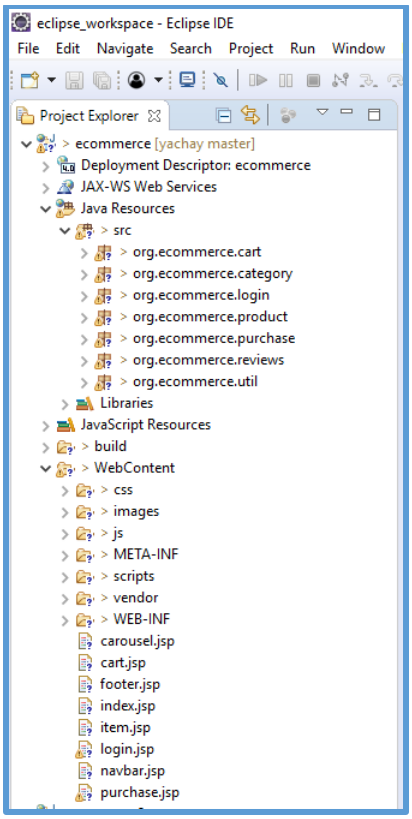
In the Import window, select the General folder, the [Projects from Folder or Archive] option and click the [Next >] button.



Click the [Directory] button, select the ecommerce folder and click the [Finish] button.

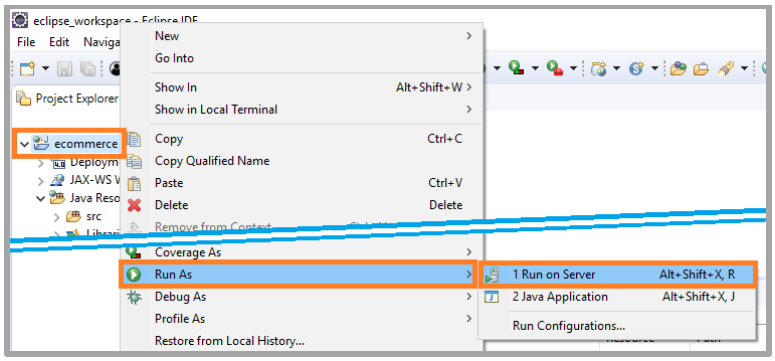


In the [Project Explorer] tab, you can see the comtents of the ecommerce eclipse project.

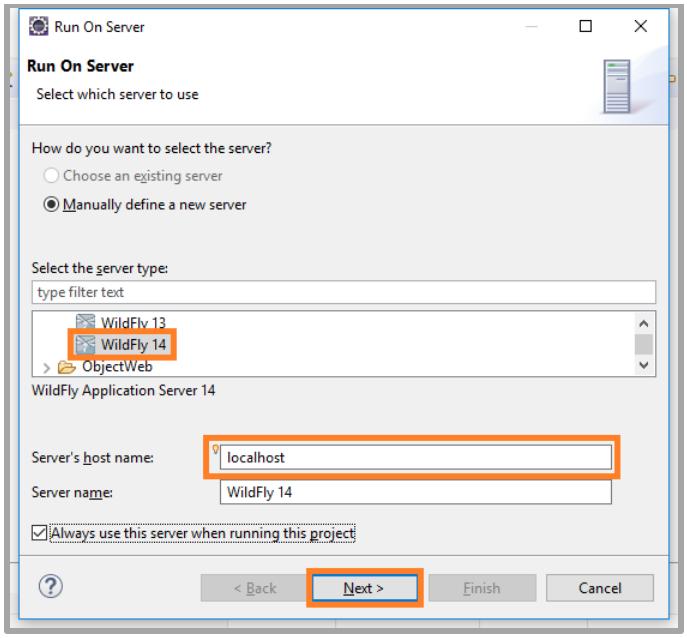


RUN THE ECOMMERCE PROJECT

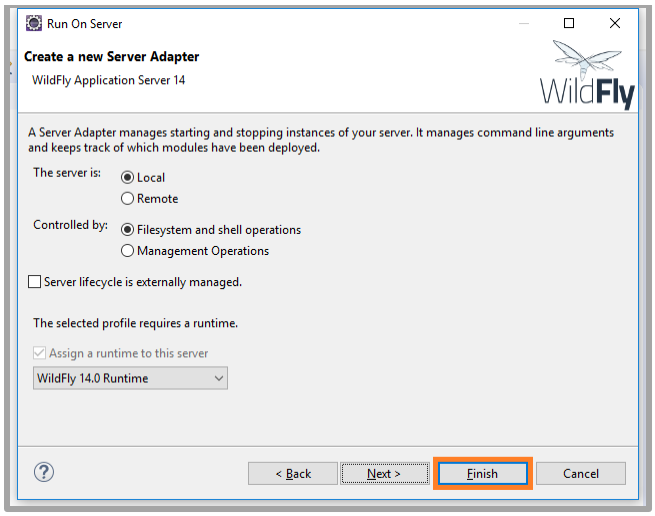
Right click the ecommerce project, select the [Run As] menu item and click the [Run on Server] option.



Select the [WildFly 14] server type and click the [Next >] button.

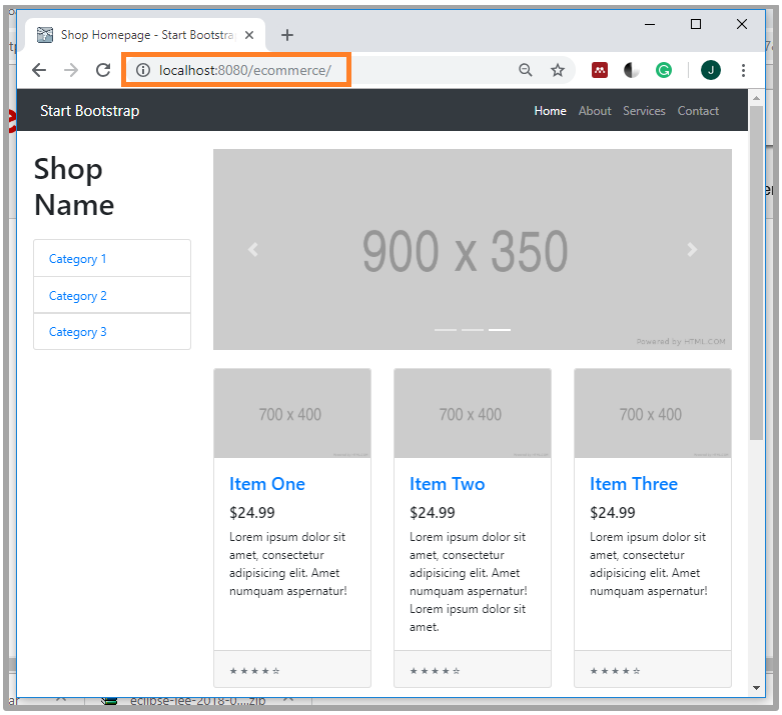


To run the server, click the [Finish] button.



To test the ecommerce web application running on the server, open a browser and write the URL.

http://localhost:8080/ecommerce/



6. REFERENCES

PyKnow

https://pyknow.readthedocs.io/en/stable/

https://github.com/buguroo/pyknow/tree/develop/docs/talks/Sistemas%20Expertos%20en%20Python%20con%20PyKnow%20-%20PyConES%202017

Turi Create

https://github.com/apple/turicreate

https://github.com/apple/turicreate/tree/master/userguide/recommender

https://pypi.org/project/turicreate/