

Class 230307

Spring - 6일차

File Upload (계속)

▼ servlet-context.xml에서 업로드 경로 설정 방법

```
<!-- 업로드 패스 설정 -->
<beans:bean class="java.lang.String" id="uploadPath">
  <beans:constructor-arg
    value="C:\develop\spring\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\CarShop\resource" />
  <!--value="C:\Develop\spring\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\BookMarket\resources" /-->
</beans:bean>

<!-- 일반 파일 업로드 경로 -->
<resources mapping="/images/**" location="/resources/images/" />
```

예외 처리

▼ @ControllerAdvice / @ExceptionHandler 사용

- 404 처리는 안됨.
- 대부분의 예외처리 방법

▼ ErrorController.java

```
package com.carshop.controller;

import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.servlet.ModelAndView;

@ControllerAdvice
public class ErrorController {

    @ExceptionHandler(RuntimeException.class)
    private ModelAndView handleErrorCommon(Exception e) {
        ModelAndView modelAndView = new ModelAndView();
        modelAndView.addObject("exception", e);
        modelAndView.setViewName("errorCommon");

        return modelAndView;
    }
}
```

▼ view.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
```

```

<body>
  ${exception}
<br>
  errorCommon.jsp
</body>
</html>

```

▼ 404 처리

▼ web.xml

```

<error-page>

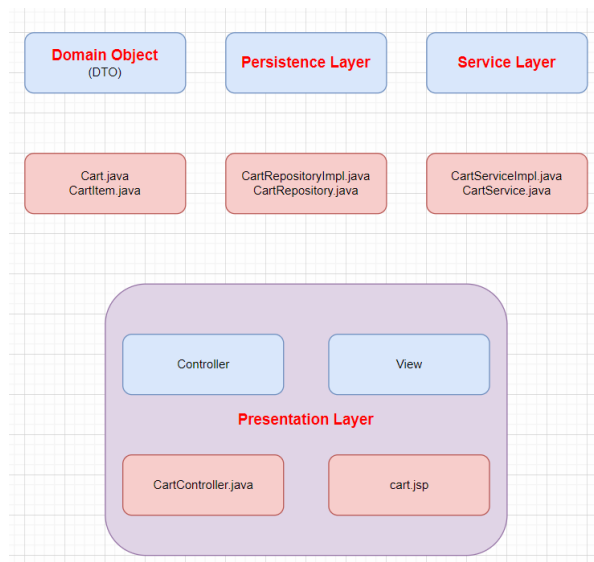
    <error-code>404</error-code>

    <location>/WEB-INF/views/error.jsp</location>

</error-page>

```

장바구니



▼ Cart.java

```

package com.carshop.controller;

import java.util.HashMap;
import java.util.Map;

public class Cart {

    private String cartId;
    private Map<String, CartItem> cartItems;
    private int grandTotal;

    public Cart() {
        cartItems = new HashMap<String, CartItem>();
    }

    public Cart(String cartId) {
        this();
        this.cartId = cartId;
    }
}

```

```

public String getCartId() {
    return cartId;
}

public void setCartId(String cartId) {
    this.cartId = cartId;
}

public Map<String, CartItem> getCartItems() {
    return cartItems;
}

public void setCartItems(Map<String, CartItem> cartItems) {
    this.cartItems = cartItems;
}

public int getGrandTotal() {
    return grandTotal;
}

public void setGrandTotal(int grandTotal) {
    this.grandTotal = grandTotal;
}

@Override
public int hashCode() {
    final int prime = 31;
    int result = 1;
    result = prime * result + ((cartId == null) ? 0 : cartId.hashCode());
    return result;
}

@Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
        return false;
    Cart other = (Cart) obj;
    if (cartId == null) {
        if (other.cartId != null)
            return false;
    } else if (!cartId.equals(other.cartId))
        return false;
    return true;
}

public void updateGrandTotal() {
    grandTotal = 0;
    for(CartItem item : cartItems.values()) {
        grandTotal = grandTotal + item.getTotalPrice();
    }
}

}

```

▼ CartRepository.java

```
package com.carshop.controller;

public interface CartRepository {

    Cart create(Cart cart);
    Cart read(String cartId);
}
```

▼ CartRepositoryImpl.java

```
package com.carshop.controller;

import java.util.HashMap;
import java.util.Map;

import org.springframework.stereotype.Repository;

@Repository
public class CartRepositoryImpl implements CartRepository {

    private Map<String, Cart> listOfCarts;

    public CartRepositoryImpl() {
        listOfCarts = new HashMap<String, Cart>();
    }

    @Override
    public Cart create(Cart cart) {
        if (listOfCarts.keySet().contains(cart.getCartId())) {
            throw new IllegalArgumentException(
                String.format("장바구니를 생성할수 없습니다.장바구니 id가 이미 존재합니다.",
                    cart.getCartId()));
        }

        listOfCarts.put(cart.getCartId(), cart);

        return cart;
    }

    @Override
    public Cart read(String cartId) {

        return listOfCarts.get(cartId);
    } // cartId 를 이용하여 장바구니에 등록된 모든 정보를 가져온다.
}
```

▼ CartServiceImpl.java

```
package com.carshop.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class CartServiceImpl implements CartService{

    @Autowired
    private CartRepository cartRepository;

    public Cart create(Cart cart) {
        return cartRepository.create(cart);
    }

    public Cart read(String cartId) {
        return cartRepository.read(cartId);
    }
}
```

▼ CartService.java

```
package com.carshop.controller;

public interface CartService {

    Cart create(Cart cart);
    Cart read(String cartId);
}
```

Python Adv - 6일차

PyQt5 사용 (계속)

▼ Combo Box

```
import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic

form_class = uic.loadUiType("mywindow6.ui")[0]

class MyWindow(QMainWindow, form_class):

    def __init__(self):
        super().__init__()
        self.setupUi(self)

        # 단추 클릭했을때 콤보 박스 확인
        #self.comboBox
        self.printitem.clicked.connect(self.printFunction)

    def printFunction(self):
        print(self.comboBox.currentText())

app = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()
```

▼ Table Widget

```
import sys
from PyQt5.QtWidgets import *

class MyWindow(QMainWindow):
    def __init__(self):
        super().__init__()

        # 윈도우 창
        self.setGeometry(800, 200, 300, 300)

        # 윈도우에 테이블위젯 넣기
        self.tableWidget = QTableWidgetItem(self)
        self.tableWidget.resize(290, 290)
        self.tableWidget.setColumnCount(2)
        self.tableWidget.setRowCount(5)

        # 테이블위젯에 데이터 넣기 (행, 열, 넣을값)
        self.tableWidget.setItem(0, 0, QTableWidgetItem("비트코인"))
        self.tableWidget.setItem(0, 1, QTableWidgetItem("31000000"))

        self.tableWidget.setItem(1, 0, QTableWidgetItem("이더리움"))
        self.tableWidget.setItem(1, 1, QTableWidgetItem("2100000"))

app = QApplication(sys.argv)
```

```
mywindow = MyWindow()  
mywindow.show()  
app.exec_()
```

크롤링과 결합해 사용

▼ 현재 온도 조회 앱

▼ ui

```
<?xml version="1.0" encoding="UTF-8"?>  
<ui version="4.0">  
  <class>MainWindow</class>  
  <widget class="QMainWindow" name="MainWindow">  
    <property name="geometry">  
      <rect>  
        <x>0</x>  
        <y>0</y>  
        <width>367</width>  
        <height>228</height>  
      </rect>  
    </property>  
    <property name="windowTitle">  
      <string>MainWindow</string>  
    </property>  
    <widget class="QWidget" name="centralwidget">  
      <widget class="QLabel" name="label">  
        <property name="geometry">  
          <rect>  
            <x>20</x>  
            <y>20</y>  
            <width>321</width>  
            <height>61</height>  
          </rect>  
        </property>  
        <property name="font">  
          <font>  
            <pointsize>21</pointsize>  
            <weight>75</weight>  
            <bold>true</bold>  
          </font>  
        </property>  
        <property name="text">  
          <string>현재 온도 조회 앱</string>  
        </property>  
        <property name="alignment">  
          <set>Qt::AlignCenter</set>  
        </property>  
      </widget>  
      <widget class="QLineEdit" name="lineEdit">  
        <property name="geometry">  
          <rect>  
            <x>40</x>  
            <y>120</y>  
            <width>131</width>  
            <height>41</height>  
          </rect>  
        </property>  
      </widget>  
      <widget class="QPushButton" name="pushButton">  
        <property name="geometry">  
          <rect>  
            <x>230</x>  
            <y>120</y>  
            <width>81</width>  
            <height>41</height>  
          </rect>  
        </property>  
        <property name="font">  
          <font>  
            <pointsize>12</pointsize>  
          </font>  
        </property>  
        <property name="text">  
          <string>조회</string>  
        </property>  
      </widget>  
    </widget>  
  </widget>  
</ui>
```

```

</widget>
</widget>
<widget class="QMenuBar" name="menubar">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>0</y>
      <width>367</width>
      <height>21</height>
    </rect>
  </property>
</widget>
<widget class="QStatusBar" name="statusbar"/>
</widget>
<resources/>
<connections/>
</ui>

```

▼ python

```

# 현재 온도 조회 앱

import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic
import requests
from bs4 import BeautifulSoup as bs

form_class = uic.loadUiType("weather.ui")[0]

class MyWindow(QMainWindow, form_class):

    def __init__(self):
        super().__init__()
        self.setupUi(self)
        # 단추를 누르면
        self.pushButton.clicked.connect(self.btnclicked)

    # 조회 후 칸에 출력
    def btnclicked(self):

        txt = requests.get('https://weather.naver.com/')
        html = bs(txt.text)
        temp = html.select('strong.current')[0].text.strip()
        result = temp[5:]

        self.lineEdit.setText(result)

app = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()

```

▼ 코인 가격 조회 앱

▼ ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>MainWindow</class>
  <widget class="QMainWindow" name="MainWindow">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>367</width>
        <height>439</height>
      </rect>
    </property>
    <property name="windowTitle">
      <string>MainWindow</string>
    </property>
    <widget class="QWidget" name="centralwidget">
      <widget class="QLabel" name="label">
        <property name="geometry">
          <rect>
            <x>20</x>

```

```

        <y>20</y>
        <width>321</width>
        <height>61</height>
    </rect>
</property>
<property name="font">
    <font>
        <pointsize>21</pointsize>
        <weight>75</weight>
        <bold>true</bold>
    </font>
</property>
<property name="text">
    <string>코인 가격 조회 앱</string>
</property>
<property name="alignment">
    <set>Qt::AlignCenter</set>
</property>
</widget>
<widget class="QPushButton" name="pushButton">
    <property name="geometry">
        <rect>
            <x>260</x>
            <y>80</y>
            <width>81</width>
            <height>41</height>
        </rect>
    </property>
    <property name="font">
        <font>
            <pointsize>12</pointsize>
        </font>
    </property>
    <property name="text">
        <string>조회</string>
    </property>
</widget>
<widget class="QTableWidget" name="tableWidget">
    <property name="geometry">
        <rect>
            <x>20</x>
            <y>150</y>
            <width>331</width>
            <height>221</height>
        </rect>
    </property>
    <property name="rowCount">
        <number>5</number>
    </property>
    <property name="columnCount">
        <number>3</number>
    </property>
    <row/>
    <row/>
    <row/>
    <row/>
    <column>
        <property name="text">
            <string>가상화폐</string>
        </property>
    </column>
    <column>
        <property name="text">
            <string>현재가격</string>
        </property>
    </column>
    <column>
        <property name="text">
            <string>비고</string>
        </property>
    </column>
</widget>
</widget>
<widget class="QMenuBar" name="menubar">
    <property name="geometry">
        <rect>
            <x>0</x>
            <y>0</y>
            <width>367</width>
            <height>21</height>
        </rect>
    </property>
</widget>

```



```

<widget class="QStatusBar" name="statusbar"/>
</widget>
<resources/>
<connections/>
</ui>

```

▼ python

```

# 코인 가격 조회 앱 (TableWidget 버전)

import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic
import pybithumb as pb

form_class = uic.loadUiType("coinwidget.ui")[0]

class MyWindow(QMainWindow, form_class):

    def __init__(self):
        super().__init__()
        self.setupUi(self)
        # 단추를 누르면
        self.pushButton.clicked.connect(self.btnclicked)

    # 조회 후 칸에 출력
    def btnclicked(self):
        tickers = ['BTC', 'ETH', 'XRP', 'ADA']

        for i, ticker in enumerate(tickers):
            price = pb.get_current_price(ticker)
            self.tableWidget.setItem(i, 0, QTableWidgetItem(ticker))
            self.tableWidget.setItem(i, 1, QTableWidgetItem(str(price)))

app = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()

```

▼ 음원 차트 조회 앱

▼ ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
<class>MainWindow</class>
<widget class="QMainWindow" name="MainWindow">
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>606</width>
<height>926</height>
</rect>
</property>
<property name="windowTitle">
<string>MainWindow</string>
</property>
<widget class="QWidget" name="centralwidget">
<widget class="QLabel" name="label">
<property name="geometry">
<rect>
<x>60</x>
<y>20</y>
<width>471</width>
<height>61</height>
</rect>
</property>
<property name="font">
<font>
<pointsize>21</pointsize>
<weight>75</weight>
<bold>true</bold>
</font>
</property>

```

[illegible]


```

        <height>21</height>
    </rect>
</property>
</widget>
<widget class="QStatusBar" name="statusbar"/>
</widget>
<resources/>
<connections/>
</ui>

```

▼ python

```

# 전체 음원 차트 조회 앱

import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic
import requests
from bs4 import BeautifulSoup as bs
from selenium import webdriver
import time

form_class = uic.loadUiType("mchart.ui")[0]

class MyWindow(QMainWindow, form_class):

    def __init__(self):
        super().__init__()
        self.setupUi(self)
        # 단추를 누르면
        self.pushButton_1.clicked.connect(self.btnClicked1) # 벅스
        self.pushButton_2.clicked.connect(self.btnClicked2) # 멜론
        self.pushButton_3.clicked.connect(self.btnClicked3) # 지니

    # 조회 후 칸에 출력
    def btnClicked1(self):

        self.label.setText(" 벅스 차트 100 ")
        self.tableWidget.clear()

        url = "https://music.bugs.co.kr/chart"
        txt = requests.get(url)
        html = bs(txt.text)
        songs = html.select('table.byChart > tbody > tr')

        for i, song in enumerate(songs):

            title = song.select('p.title > a')[0].text
            singer = song.select('p.artist > a')[0].text

            self.tableWidget.setItem(i, 0, QTableWidgetItem("Bugs"))
            self.tableWidget.setItem(i, 1, QTableWidgetItem(title))
            self.tableWidget.setItem(i, 2, QTableWidgetItem(singer))

    def btnClicked2(self):

        self.label.setText(" 멜론 차트 100 ")
        self.tableWidget.clear()

        driver = webdriver.Chrome('chromedriver.exe')
        driver.get("https://www.melon.com/chart/index.htm")
        txt = driver.page_source
        html = bs(txt)

        songs = html.select('tbody > tr')

        for i, song in enumerate(songs):
            title = song.select('div.rank01> span > a')[0].text
            singer = song.select('div.rank02> a')[0].text

            self.tableWidget.setItem(i, 0, QTableWidgetItem("Melon"))
            self.tableWidget.setItem(i, 1, QTableWidgetItem(title))
            self.tableWidget.setItem(i, 2, QTableWidgetItem(singer))

    def btnClicked3(self):

        self.label.setText(" 지니 차트 100 ")
        self.tableWidget.clear()

        for p in range(0, 2):
            driver = webdriver.Chrome('chromedriver.exe') # 버전 주의

```

```

url = 'https://www.genie.co.kr/chart/top200?pg=' + str(p+1)
driver.get(url)
txt = driver.page_source
html = bs(txt)

songs = html.select('tbody > tr')

for i, song in enumerate(songs):
    title = song.select('a.title')[0].text.strip()
    singer = song.select('a.artist')[0].text.strip()

    self.tableWidget.setItem(i + (p*50), 0, QTableWidgetItem("Genie"))
    self.tableWidget.setItem(i + (p*50), 1, QTableWidgetItem(title))
    self.tableWidget.setItem(i + (p*50), 2, QTableWidgetItem(singer))

    time.sleep(1)

app = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()

```