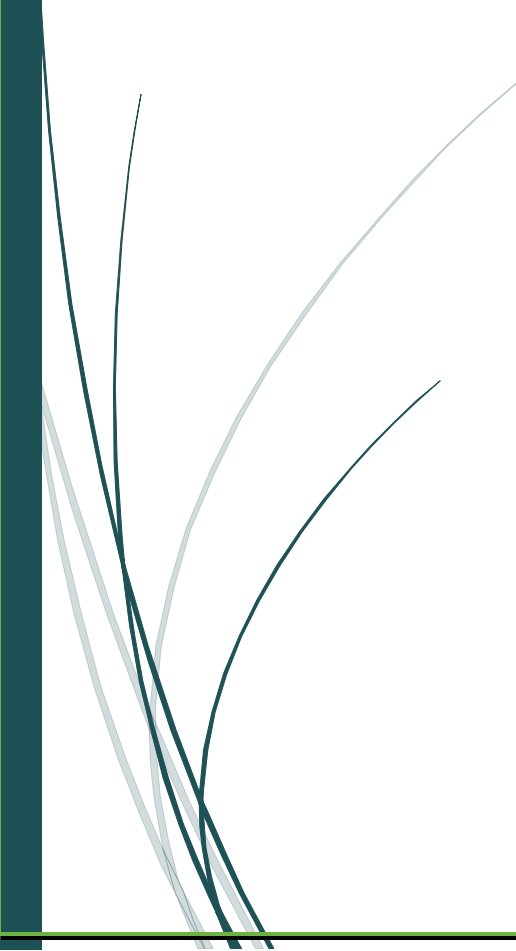


The logo for the Spring framework, featuring a red arrow pointing to the right with the word "Spring" written inside in white.

Spring

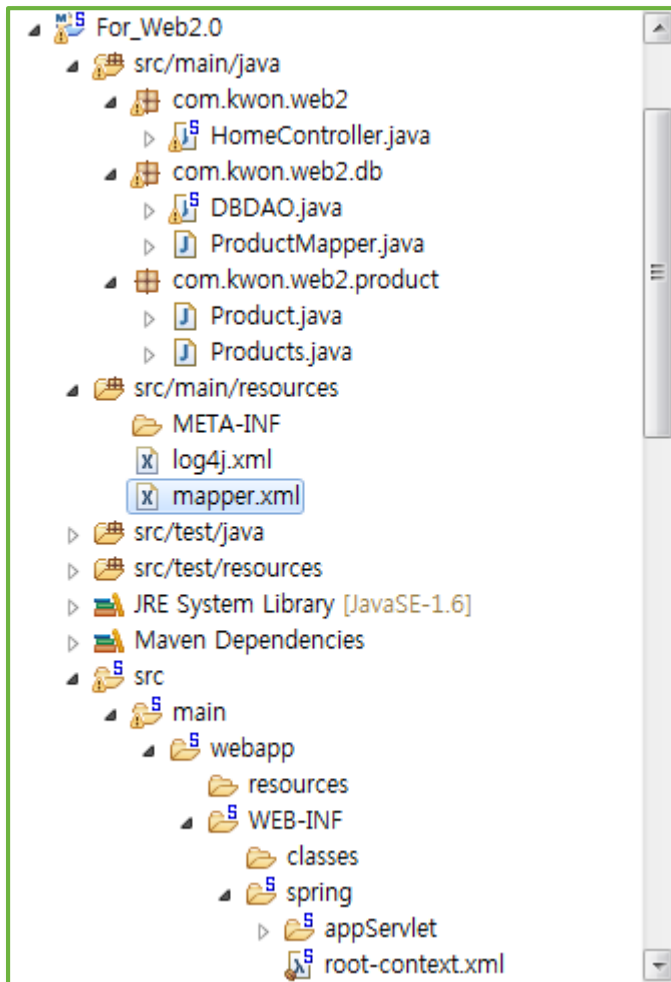
For Web2.0

A decorative graphic in the bottom-left corner consisting of several thin, curved lines in dark blue and light grey, resembling stylized grass or reeds.

권기웅

1.XML 응답하기

1.1 프로젝트 전체 구조



1.2 mapper

1.2.1 mapper.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.kwon.web2.db.ProductMapper">
    <select id="getProduct" resultType="com.kwon.web2.product.Product">
        select *
        from jun11_product_kwon
        order by p_name
    </select>
</mapper>
```

1.2.2 ProductMapper.java

```
public interface ProductMapper {
    public abstract List<Product> getProduct();
}
```



1.3 JavaBean

1.3.1 Product.java

```
@XmlRootElement(name = "product")
public class Product {
    private String p_id;
    private String p_name;
    private BigDecimal p_price;

    public Product() {}

    public String getP_id() {return p_id;}

    @XmlElement
    public void setP_id(String p_id) {this.p_id = p_id;}

    public String getP_name() {return p_name;}

    @XmlElement
    public void setP_name(String p_name) {this.p_name = p_name;}

    public BigDecimal getP_price() {return p_price;}

    @XmlElement
    public void setP_price(BigDecimal p_price) {this.p_price = p_price;}
}
```



1.3.2 Products.java

```
@XmlElement(name = "products")
public class Products {
    private List<Product> products;

    public Products() {}

    public Products(List<Product> products) {
        super();
        this.products = products;
    }

    public List<Product> getProducts() {return products;}

    @XmlElement
    public void setProducts(List<Product> products) {this.products = products;}
}
```

1.4 DBDAO

1.4.1 DBDAO.java

```
@Service
public class DBDAO {

    @Autowired
    private SqlSession ss;

    public Products getProduct() {
        ProductMapper pm = ss.getMapper(ProductMapper.class);
        return new Products(pm.getProduct());
    }
}
```



1.5 Controller

1.5.1 HomeController.java

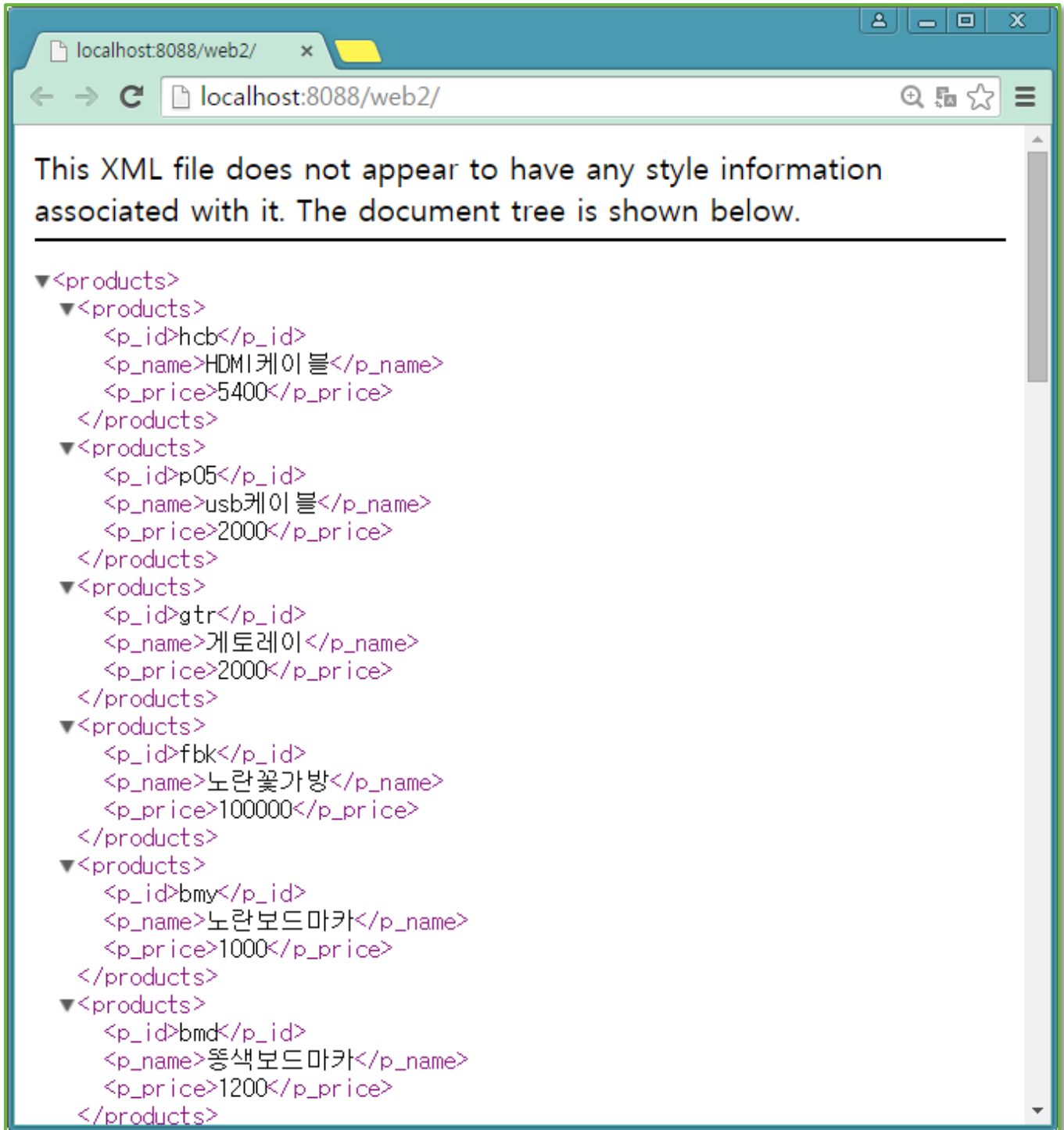
```
@Controller
public class HomeController {

    @Autowired
    private DBDAO d;

    @RequestMapping(value = "/", method = RequestMethod.GET)
    public @ResponseBody Products getXML() {
        return d.getProduct();
    }
}
```



1.6 실행결과



2.JSON 응답하기

2.1 Maven 으로 jackson-mapper-asl.jar 추가

```
<dependency>
  <groupId>org.codehaus.jackson</groupId>
  <artifactId>jackson-mapper-asl</artifactId>
  <version>1.9.13</version>
</dependency>
```

2.2 src/main/webapp/WEB-INF/spring/root-context.xml 에 jackson 추가

```
<beans:bean id="jackson" class="org.springframework.http.converter.json.MappingJacksonHttpMessageConverter">
</beans:bean>
```

2.3 JavaBean 에 @XML~~~ 삭제

2.3.1 Product.java

```
public class Product {
    private String p_id;
    private String p_name;
    private BigDecimal p_price;

    public Product() {}

    public String getP_id() {return p_id;}
    public void setP_id(String p_id) {this.p_id = p_id;}
    public String getP_name() {return p_name;}
    public void setP_name(String p_name) {this.p_name = p_name;}
    public BigDecimal getP_price() {return p_price;}
    public void setP_price(BigDecimal p_price) {this.p_price = p_price;}
}
```



2.3.2 Products.java

```
public class Products {  
    private List<Product> products;  
  
    public Products() {}  
  
    public Products(List<Product> products) {  
        super();  
        this.products = products;  
    }  
  
    public List<Product> getProducts() {return products;}  
    public void setProducts(List<Product> products) {this.products = products;}  
}
```

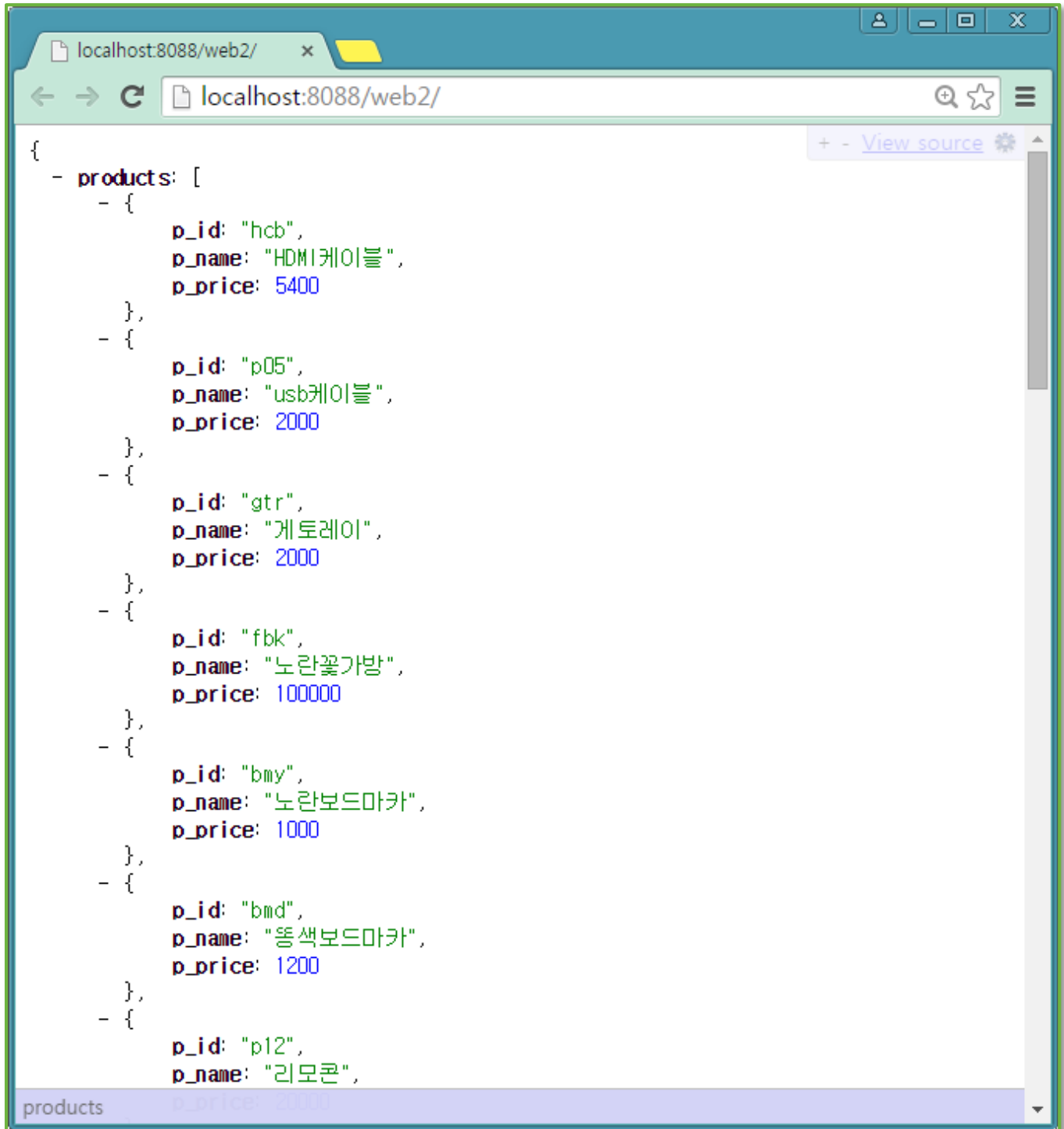
2.4 Controller

2.4.1 HomeController.java

```
@Controller  
public class HomeController {  
  
    @Autowired  
    private DBDAO d;  
  
    @RequestMapping(value = "/", method = RequestMethod.GET)  
    public @ResponseBody Products getJSON() {  
        return d.getProduct();  
    }  
}
```



2.5 실행결과



The screenshot shows a web browser window with the address bar displaying 'localhost:8088/web2/'. The main content area displays a JSON array of product objects. The JSON is formatted with syntax highlighting: property names are in black, string values are in green, and numeric values are in blue. A 'View source' link is visible in the top right corner of the content area. The browser's status bar at the bottom shows 'products' and 'p_price: 2000'.

```
{
  - products: [
    - {
      p_id: "hcb",
      p_name: "HDMI케이블",
      p_price: 5400
    },
    - {
      p_id: "p05",
      p_name: "usb케이블",
      p_price: 2000
    },
    - {
      p_id: "gtr",
      p_name: "게토레이",
      p_price: 2000
    },
    - {
      p_id: "fbk",
      p_name: "노란꽃가방",
      p_price: 100000
    },
    - {
      p_id: "bmy",
      p_name: "노란보드마카",
      p_price: 1000
    },
    - {
      p_id: "bmd",
      p_name: "뽕색보드마카",
      p_price: 1200
    },
    - {
      p_id: "p12",
      p_name: "리모콘",
      p_price: 2000
    }
  ]
}
```



2.6 JSONP

2.6.1 HomeController.java 에 추가

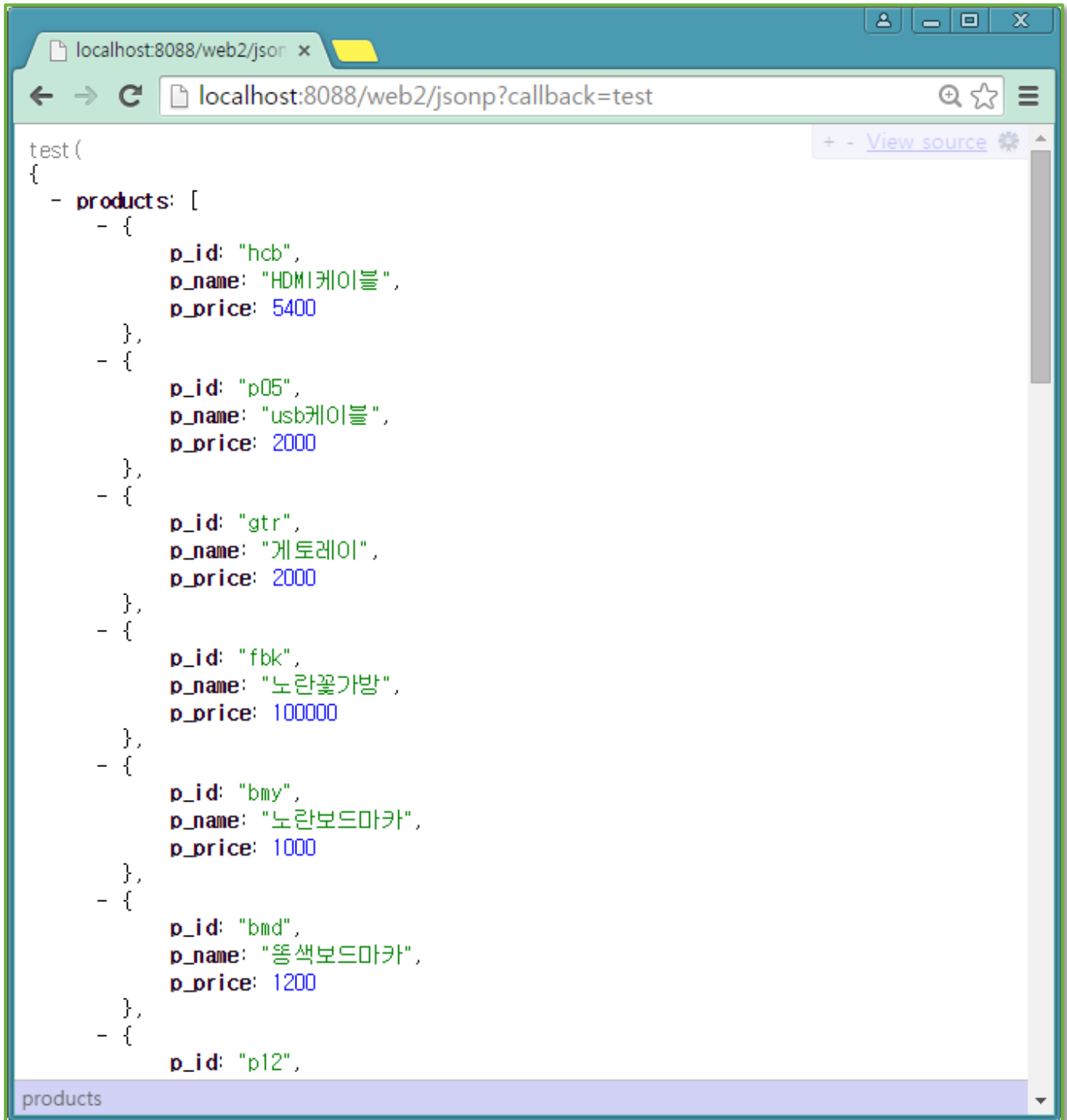
```
@RequestMapping(value = "/jsonp", method = RequestMethod.GET)
public @ResponseBody String getJSONP(
    @RequestParam(value = "callback") String cb) {

    try {
        Products p = d.getProduct();
        ObjectMapper om = new ObjectMapper();
        String res = cb + "(" + new String(om.writeValueAsBytes(p), "8859_1") + ")";

        return res;
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        return null;
    }
}
```



2.6.2 실행결과



A screenshot of a web browser window. The address bar shows the URL `localhost:8088/web2/jsonp?callback=test`. The page content displays a JSON object with a `products` array. The array contains five product objects, each with `p_id`, `p_name`, and `p_price` fields. The products are: 'hcb' (HDMI케이블, 5400), 'p05' (usb케이블, 2000), 'gtr' (게토레이, 2000), 'fbk' (노란꽃가방, 100000), and 'bmy' (노란보드마카, 1000). The JSON is partially visible, showing the first five products and the start of the sixth. A 'View source' link is visible in the top right corner of the content area.

```
test (
{
  - products: [
    - {
      p_id: "hcb",
      p_name: "HDMI케이블",
      p_price: 5400
    },
    - {
      p_id: "p05",
      p_name: "usb케이블",
      p_price: 2000
    },
    - {
      p_id: "gtr",
      p_name: "게토레이",
      p_price: 2000
    },
    - {
      p_id: "fbk",
      p_name: "노란꽃가방",
      p_price: 100000
    },
    - {
      p_id: "bmy",
      p_name: "노란보드마카",
      p_price: 1000
    },
    - {
      p_id: "bmd",
      p_name: "뽕색보드마카",
      p_price: 1200
    },
    - {
      p_id: "p12",
```

products

