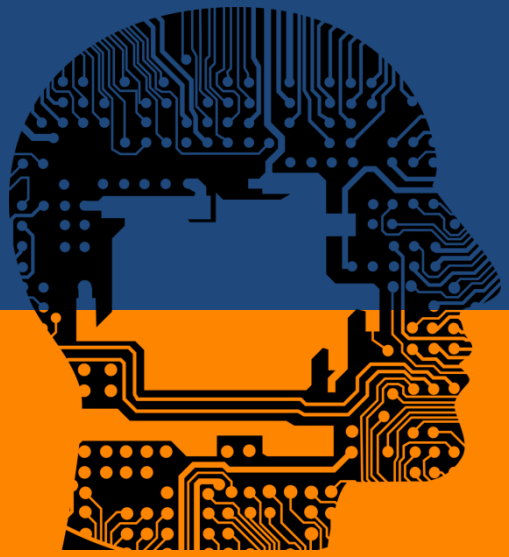
한국IT진흥부설
정보보호교육학원 아이섹

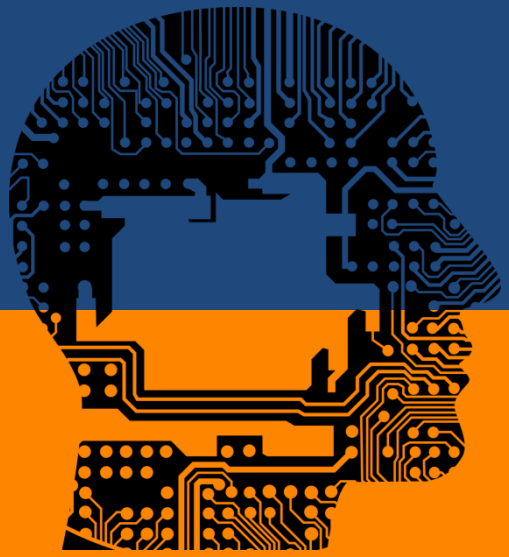


JAVA 웹 개발자 양성과정
Spring: Spring MVC

2강 - 서블릿

By SoonGu Hong

한국IT진흥부설
정보보호교육학원 아이섹



JAVA 웹 개발자 양성과정

Spring: Spring MVC

1. 프로젝트 생성

스프링 부트 서블릿 환경 구성

`@ServletComponentScan`

스프링 부트는 서블릿을 직접 등록해서 사용할 수 있도록 `@ServletComponentScan`을 지원한다. 다음과 같이 추가하자.

hello.servlet.ServletApplication

```
package hello.servlet;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.boot.web.servlet.ServletComponentScan;

@ServletComponentScan //서블릿 자동 등록
@SpringBootApplication
public class ServletApplication {

  public static void main(String[] args) {
      SpringApplication.run(ServletApplication.class, args);
  }
```

서블릿 등록하기

처음으로 실제 동작하는 서블릿 코드를 등록해보자.

```
@WebServlet(name = "helloServlet", urlPatterns = "/hello")
public class HelloServlet extends HttpServlet {

    @Override
    protected void service(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {

        System.out.println("HelloServlet.service");
        System.out.println("request = " + request);
        System.out.println("response = " + response);

        String username = request.getParameter("username");
        System.out.println("username = " + username);

        response.setContentType("text/plain");
        response.setCharacterEncoding("utf-8");
        response.getWriter().write("hello " + username);
    }
}
```

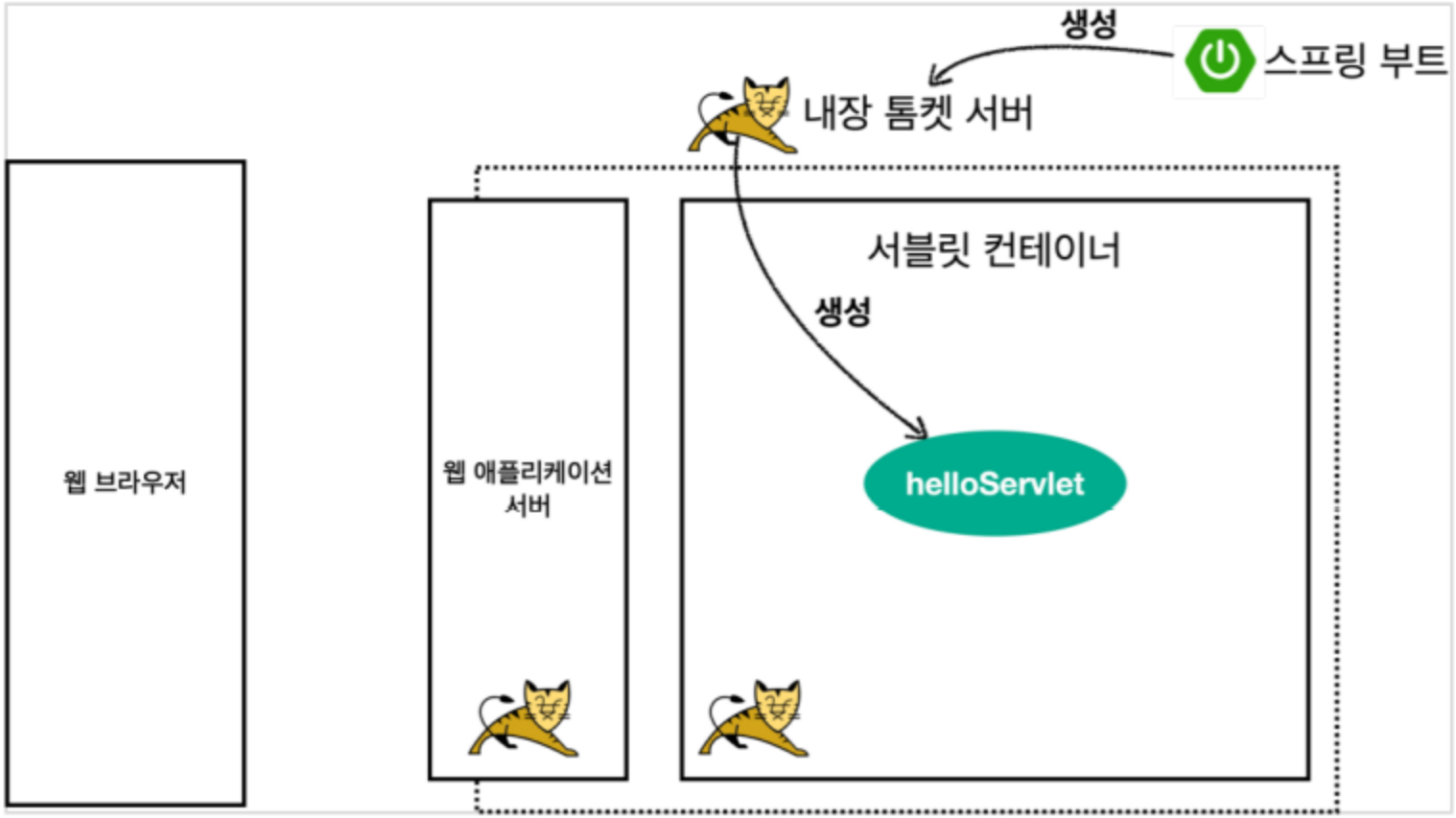
- `@WebServlet` 서블릿 애노테이션
  - name: 서블릿 이름
  - urlPatterns: URL 매핑

HTTP 요청을 통해 매핑된 URL이 호출되면 서블릿 컨테이너는 다음 메서드를 실행한다.

```
protected void service(HttpServletRequest request, HttpServletResponse response)
```

- 웹 브라우저 실행
  - http://localhost:8080/hello?username=world
  - 결과: hello world
- 콘솔 실행결과

```
HelloServlet.service
request = org.apache.catalina.connector.RequestFacade@5e4e72
response = org.apache.catalina.connector.ResponseFacade@37d112b6
username = world
```


생성
스프링 부트
내장 톰켓 서버
웹 브라우저
웹 애플리케이션 서버
서블릿 컨테이너
생성
helloServlet

HTTP 요청, HTTP 응답 메시지

[HTTP 요청]

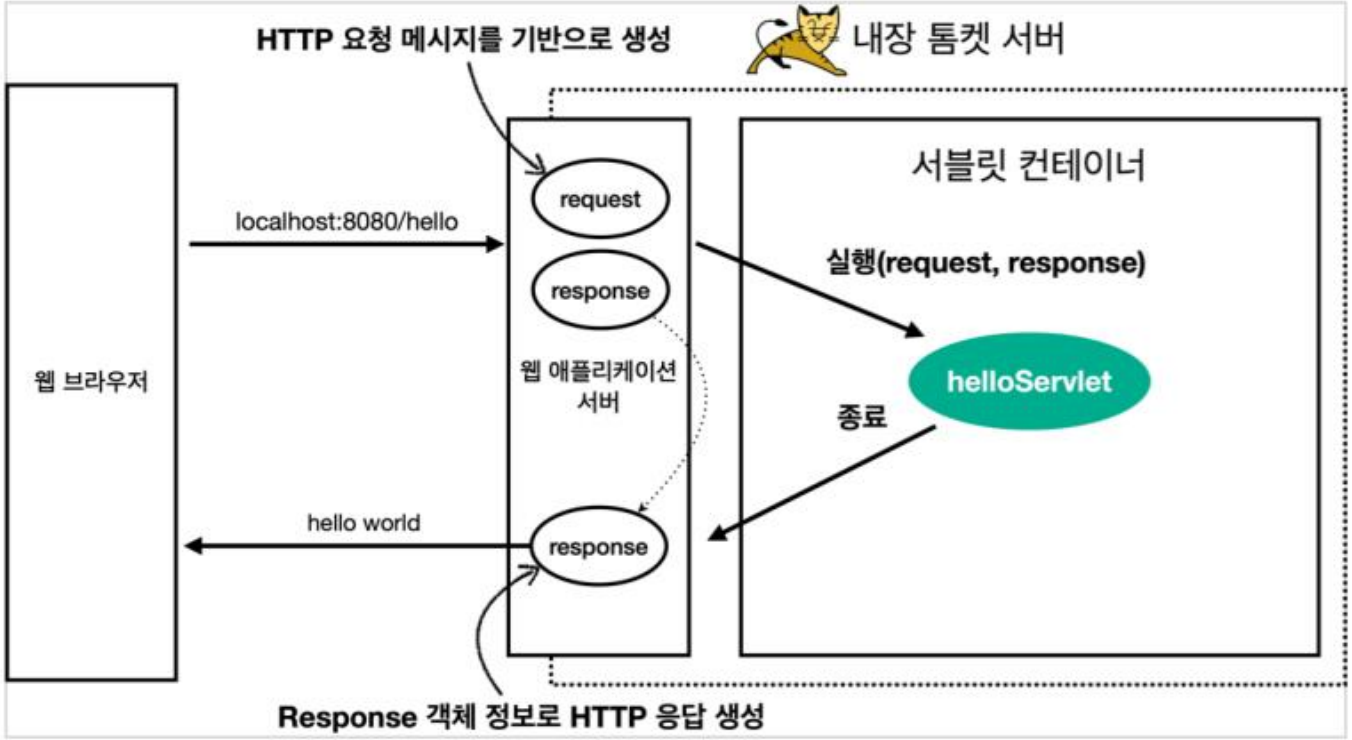
```
GET /hello?username=world HTTP/1.1
Host: localhost:8080
```

[HTTP 응답]

```
HTTP/1.1 200 OK
Content-Type: text/plain;charset=utf-8
Content-Length: 11

hello world
```

웹 애플리케이션 서버의 요청 응답 구조



welcome 페이지 추가

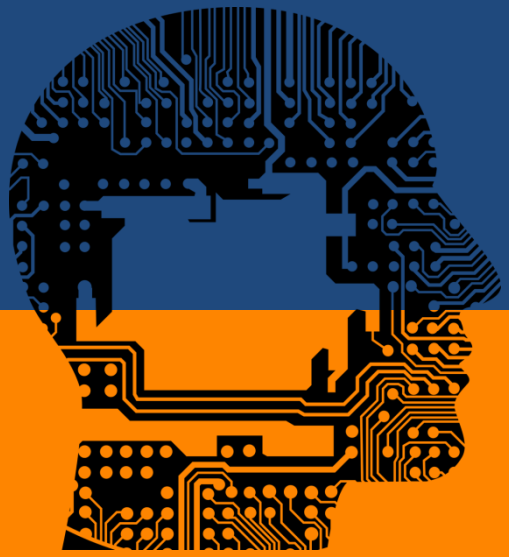
지금부터 개발할 내용을 편리하게 참고할 수 있도록 welcome 페이지를 만들어두자.

`webapp` 경로에 `index.html` 을 두면 http://localhost:8080 호출시 `index.html` 페이지가 열린다.

`main/webapp/index.html`

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<ul>
    <li><a href="basic.html">서블릿 basic</a></li>
</ul>
</body>
</html>
```

한국IT진흥부설
정보보호교육학원 아이섹



JAVA 웹 개발자 양성과정

Spring: Spring MVC

2. HttpServletRequest

HttpServletRequest 역할

HTTP 요청 메시지를 개발자가 직접 파싱해서 사용해도 되지만, 매우 불편할 것이다. 서블릿은 개발자가 HTTP 요청 메시지를 편리하게 사용할 수 있도록 개발자 대신에 HTTP 요청 메시지를 파싱한다. 그리고 그 결과를 `HttpServletRequest` 객체에 담아서 제공한다.

HttpServletRequest를 사용하면 다음과 같은 HTTP 요청 메시지를 편리하게 조회할 수 있다.

HTTP 요청 메시지

```
POST /save HTTP/1.1
Host: localhost:8080
Content-Type: application/x-www-form-urlencoded

username=kim&age=20
```

임시 저장소 기능

- 해당 HTTP 요청이 시작부터 끝날 때 까지 유지되는 임시 저장소 기능
  - 저장: `request.setAttribute(name, value)`
  - 조회: `request.getAttribute(name)`

세션 관리 기능

- `request.getSession(create: true)`

HTTP 요청 데이터 - 개요

HTTP 요청 메시지를 통해 클라이언트에서 서버로 데이터를 전달하는 방법을 알아보자.

주로 다음 3가지 방법을 사용한다.

- **GET - 쿼리 파라미터**
  - /url**?username=hello&age=20**
  - 메시지 바디 없이, URL의 쿼리 파라미터에 데이터를 포함해서 전달
  - 예) 검색, 필터, 페이징등에서 많이 사용하는 방식
- **POST - HTML Form**
  - content-type: application/x-www-form-urlencoded
  - 메시지 바디에 쿼리 파리미터 형식으로 전달 username=hello&age=20
  - 예) 회원 가입, 상품 주문, HTML Form 사용
- **HTTP message body**에 데이터를 직접 담아서 요청
  - HTTP API에서 주로 사용, JSON, XML, TEXT
- 데이터 형식은 주로 JSON 사용
  - POST, PUT, PATCH

HTTP 요청 데이터 - GET 쿼리 파라미터

다음 데이터를 클라이언트에서 서버로 전송해보자.

전달 데이터

- username=hello
- age=20

메시지 바디 없이, URL의 **쿼리 파라미터**를 사용해서 데이터를 전달하자.
예) 검색, 필터, 페이징등에서 많이 사용하는 방식

쿼리 파라미터는 URL에 다음과 같이 `?` 를 시작으로 보낼 수 있다. 추가 파라미터는 `&` 로 구분하면 된다.

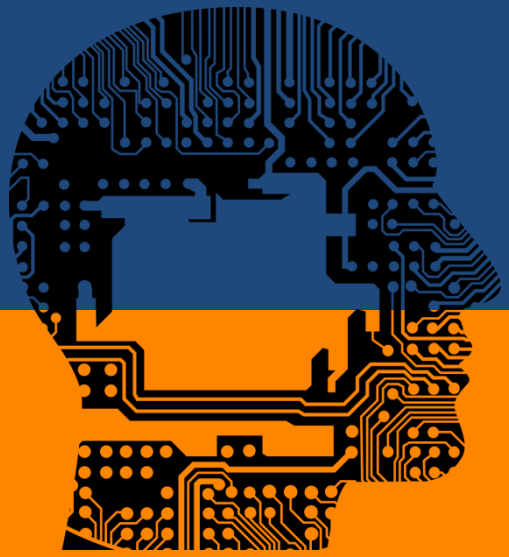
- http://localhost:8080/request-param?username=hello&age=20

서버에서는 `HttpServletRequest` 가 제공하는 다음 메서드를 통해 쿼리 파라미터를 편리하게 조회할 수 있다.

쿼리 파라미터 조회 메서드

```
String username = request.getParameter("username"); //단일 파라미터 조회
Enumeration<String> parameterNames = request.getParameterNames(); //파라미터 이름들 모두 조회
Map<String, String[]> parameterMap = request.getParameterMap(); //파라미터를 Map 으로 조회
String[] usernames = request.getParameterValues("username"); //복수 파라미터 조회
```

한국IT진흥부설
정보보호교육학원 아이섹



JAVA 웹 개발자 양성과정

Spring: Spring MVC

3. HttpServletResponse

HttpServletResponse - 기본 사용법

HttpServletResponse 역할

HTTP 응답 메시지 생성

- HTTP 응답코드 지정
- 헤더 생성
- 바디 생성

편의 기능 제공

- Content-Type, 쿠키, Redirect

감사합니다
THANK YOU