

# Zero\_to\_hero

2025. március 30., vasárnap 18:05

Entity:

```
public class Pet {
```

Private fieldek

```
private Long id;
```

Több konstruktor is lehetséges

```
public Pet(Long id, String name, String chipCode, String address, String species) {
    this.id = id;
    this.name = name;
    this.chipCode = chipCode;
    this.address = address;
    this.species = species;
}
```

Getter + setter

hashCode, equals, toString

PetRepository - 1 példány létezen, ezért:

```
private static final PetRepository PET_REPOSITORY = new PetRepository();
```

```
public static PetRepository getInstance() {
    return PET_REPOSITORY;
}
```

Ilyenkor konstruktor privát!! Ez a singleton módszer, így érhetjük el, hogy csak 1 példány létezen az egész appban, így nem lehet konstruktoron keresztül inicialni

Először: jsp elkészítés

Táblázat style:

```
<style>
td {
    border: 1px solid black;
}
</style>
```

Taglibs:

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
```

mindegyikhez

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

forEachhez, ifhez stb.

```
<%@ taglib prefix="t" tagdir="/WEB-INF/tags" %>
```

```
<%@ taglib prefix="f" uri="/WEB-INF/fds/favorite.tld" %>
```

```
<t:page title="Phone List">
```

```
<c:forEach items="${phoneList}" var="phone">
```

phoneList -> ListServlet

@Override

```
public void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    req.setAttribute("phoneList", phoneService.findAll());
}
```

**Servletek:**

doGet, doPost

doGet: mit jelenítsen meg, doPost: post után mi történjen

Pl.

```
public class MainServlet extends HttpServlet {
```

@Override

```
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException, ServletException {
    req.getRequestDispatcher("mainPage.jsp").forward(req, resp);
}
```

mainPage.jsp-t kapja meg:

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
```

```
<%@ taglib prefix="mytag" tagdir="/WEB-INF/tags" %>
```

```
<mytag:page title="Pet clinic" />
```

Ehhez kell WEB-INF-ben egy tags mappa, azon belül pedig létrehozni egy xy.tag fájlt

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ tag dynamic-attributes="params" language="java" pageEncoding="UTF-8" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html" charset="UTF-8">
<meta charset="UTF-8">
```

Ha nem használok custom jspt- akkor ezt a 2 meta cuccost az összes jsp-be be kell írni!!!

```
<title>${params.get("title")}</title>
</head>
<body>
<h1>Welcome here!</h1>
<div>
<div>
<a href="${pageContext.request.contextPath}/addPetServlet">Add new pet</a>
</div>
<div>
<a href="${pageContext.request.contextPath}/listServlet">Pet list</a>
</div>
</div>
<div>
<jsp:doBody/>
</div>
</body>
</html>
```

Többi jspben:

```
<%@ page contentType="text/html; charset=UTF-8" language="java" pageEncoding="UTF-8" %>
```

```
<%@ taglib prefix="t" tagdir="/WEB-INF/tags" %>
```

```
<t:page title="Add new pet">
```

Ez mehet mainPage-jspe is, ha akarok ilyet

Oldalak neve (title):

Xy.tag fájl:

```
<%@ tag dynamic-attributes="params" language="java" pageEncoding="UTF-8" %>
```

..

```
<title>${params.get("title")}</title>
```

Többi jps fájl:

```
<%@ taglib prefix="t" tagdir="/WEB-INF/tags" %>
```

<t:page title="Add new pet">

így megkapja a title-t  
</t:page> a végén!!!

Fontos, hogy ha valamit submitolni akarok jsbpen, akkor <form method="post"> ... </form> közé kell zárni a dolgokat!

```
req.setCharacterEncoding("UTF-8");
Mikor kell?
- POST requestekben
- getParameter előtt

@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException, ServletException {
    req.setAttribute("petList", petService.findAll());
    req.getRequestDispatcher("petList.jsp").forward(req, resp);
}
```

Mi történik?

- Servlet kap egy GET kérést
- petService.findAll() adatot gyűjt és petList nevű attribútumban tárolja
- getRequestDispatchernek átadjuk, hogy hova küldje az előbb letárolt adatokat, és forward-dal átküldjük

Gomb, checkbox vagy bármilyen hozzákötése a entity egy tulajdonságához hidden input type használatával (id) :

```
<td>
    <form method="post" action="listServlet">
        <button type="submit" name="action" value="delete">Delete</button>
        <input type="hidden" name="petId" value="${pet.id}">
    </form>
</td>
```

Itt a Delete gombhoz kötjük a pet.id-t, de elrejtjük az id-t

```
<groupId>an2myc</groupId>
<artifactId>PetClinic</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>war</packaging>
```

Url: PetClinic\_war/servletNeve lesz így

Petclinic\_war ez miatt a pom.xmlben lévő miatt az

Favorite taglib:

```
<tlb-version>1.0.0</tlb-version>
<short-name>FavoriteTag</short-name>
<tag>
    <name>favoriteTag</name>
    <tag-class>an2myc.tag.FavoritePhoneTag</tag-class>
    <body-content>empty</body-content>
</tag>
</taglib>
```

getGlobalID taglib:

```
<tlb-version>1.0.0</tlb-version>
<short-name>GlobalId</short-name>
<tag>
    <name>globalIdTag</name>
    <tag-class>mik.prog4.zh1.tag.GlobalIdentifierTag</tag-class>
    <body-content>empty</body-content>
    <attribute>
        <name>pet</name>
        <required>true</required>
        <rtexprvalue>true</rtexprvalue>
    </attribute>
</tag>
</taglib>
```

Ezeknek használata:

- Ahol meg akarom hogy jelenjen oda illesztsem ezt:

```
<f:favoriteTag/>
```

f: a prefix amit megadtam

favoriteTag: webapp -> WEB-INF -> tlds -> favorite.tld és TAG NAME

mvn compile -> build success legyen

Project létrehozásánál create git repository?

Custom tag kiírás: JspWriter

```
@Override
public void doTag() throws IOException {
    JspWriter out = getJspContext().getOut();
    if (pet != null) {
        String globalIdentifier = String.valueOf(petService.getGlobalIdentifier(pet));
        out.println(String.format(globalIdentifier));
    }
}
```

```
<td><g:globalIdTag pet="${pet}" /></td>
```

Itt a pet a paraméter amit megkap a custom tag, ahhoz hogy ezt meg tudjuk csinálni, a tag.tld -ben ez kell:

```
<attribute>
    <name>pet</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
</attribute>
```

```
public class GlobalIdentifierTag extends SimpleTagSupport {
```

```
    private Pet pet;
    private static final PetService petService = new PetService();
```

```
    public void setPet(Pet pet) {
        this.pet = pet;
    }
```

```
req.setAttribute("id", id);
```

```
String id = Optional.ofNullable(req.getCookies())
    .map(Stream::of)
    .orElseGet(Stream::empty)
    .filter(cookie -> "id".equals(cookie.getName()))
    .findFirst()
    .map(cookie -> URLDecoder.decode(cookie.getValue()))
    .orElse(null);
```

```
// Static inner helper class responsible for holding the singleton instance private static class Holder { private static final PhoneRepository INSTANCE = new PhoneRepository(); }
```

```
Req.getRequestDispatcher.jsp-re mutat
```

```

Resp.sendRedirect(req.getContextPath()+"/servletNeve");
Req.setCharacterEncoding doPost-ban és getParameter előtt

Cookie: name + URLEncoder.encode(xy, "UTF-8");
Cookie responsehoz adás: resp.addCookie(name)
Req.setAttribute("id", selectedPhone); ez azért hogy ne kelljen 2x kattintani

Tag:
doTag method

```

#### Multiple Cookies in a Request:

- When a client (browser) sends an HTTP request to your server, all cookies associated with your domain are sent together in the request. These cookies are provided as an array (Cookie[]) in the HttpServletRequest object.
- This array could include:
- Cookies you explicitly set (e.g., the phone ID cookie).
- Default cookies like JSESSIONID (used for session tracking).

```

var context = (PageContext) this.getJspContext();
var request = (HttpServletRequest) context.getRequest();

JspWriter out = this.getJspContext().getOut();

String phonelid = request.getParameter("id");
if (phonelid == null) {
    phonelid = getCookieValue(request.getCookies(), "id");
}

if (phonelid != null) {
    var phone = phoneService.findById(Long.parseLong(phonelid));
    out.println(phone != null ? "Favorite phone: " + phone.getManufacturer() + " " + phone.getType() + " " + phone.getimei() : "You don't have a favorite phone");
}

private String getCookieValue(Cookie[] cookies, String cookieName) {
    if (cookies != null) {
        for (Cookie cookie : cookies) {
            if (cookieName.equals(cookie.getName())) {
                return cookie.getValue();
            }
        }
    }
    return null;
}

```

- A Java alapvetően biztonságosan kezeli a null értékeket addig, amíg nem próbálsz meg rajtuk műveleteket végrehajtani.

<https://grape-nail-f1d.notion.site/prog-4-zh-ra-1c7af4e1577780abbc40d76d9f6d7c9a>

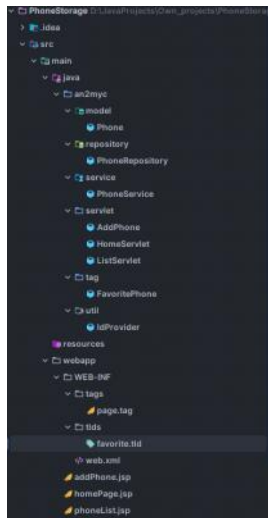
```

<groupId>an2myc</groupId>
<artifactId>PetClinic</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>war</packaging>

```

Url: PetClinic\_war/mainServlet

Petclinic\_war ez miatt a pom.xmlben lévő miatt az



```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="t" tagdir="/WEB-INF/tags" %>
<%@ taglib prefix="f" uri="/WEB-INF/tlds/favorite.tld" %>

```

```

<style>
td {
    border: 1px solid black;

```

}  
</style>