

JAVA WEB PROGRAMIRANJE

Thymeleaf Predlošci

Napredne vrste predložaka, žičani okviri i primjeri koda

HTML Predlošci

XML Predlošci

Tekstualni predlošci

JS i CSS predlošci

E-mail predlošci

Fragment predlošci

Sigurnosna integracija

Neobrađeni (RAW) predlošci

Teme kolegija

Što ćemo obraditi danas

01

Uvod u Thymeleaf

Pregled pogona predložaka i postavljanje

02

HTML Thymeleaf predlošci

Dinamična HTML generacija s th: atributima

03

XML Thymeleaf predlošci

XML odgovori REST API-ja

04

Tekstualni predlošci

Generiranje prostog teksta, e-maila i koda

05

JavaScript i CSS predlošci

Integracija statičkih resursa i inline izrazi

06

Fragment predlošci

Komponente rasporeda koje se mogu ponovo koristiti

07

E-mail predlošci

Generiranje HTML e-maila sa Spring Mail

08

Sigurnost i i18n integracija

Thymeleaf Security dijalekt i internacionalizacija

09

Neobrađeni (RAW) predlošci

Izlazni način s minimalnom obradom

Uvod u Thymeleaf

Serverski Java pogon predložaka za web i samostalna okruženja

Načini predložaka

HTML	Web stranice — najčešći način
XML	Dobro oblikovani XML dokumenti
TEXT	Prosti tekst, e-mail, izlaz koda
JAVASCRIPT	Dinamično generiranje JS datoteka
CSS	Dinamično generiranje stilova
RAW	Neobrađeni prolazni izlaz

Ključne značajke

- Prirodni predlošci — rade bez poslužitelja
- Duboka integracija sa Spring Bootom
- Sustav dijalekata za prilagođene attribute
- Zadana mapa: `src/main/resources/templates`
- Ovisnost: `spring-boot-starter-thymeleaf`
- Prostor naziva: `xmlns:th="http://www.thymeleaf.org"`

Zadana lokacija: `src/main/resources/templates/`

01 — HTML Thymeleaf predlošci

Dinamično generiranje HTML-a na poslužitelju, vraća čisti HTML pregledniku

Žičani okvir

 http://localhost:8080/studenti

Popis studenata

IT001	Ana Kovač	A
IT002	Marko Horvat	B
IT003	Petra Šimić	A+

Dodaj studenta

Kod predloška

```
<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<body>

<h1 th:text="${naslov}">Naslov</h1>

<table>

  <tr th:each="s : ${studenti}">

    <td th:text="${s.id}">id</td>

    <td th:text="${s.ime}">ime</td>

    <td th:text="${s.ocjena}">ocjena</td>

  </tr>

</table>

</body></html>
```

02 — XML Thymeleaf predlošci

Vraćanje strukturiranih XML podataka iz REST kontrolera — korisno za sistemske integracije

pom.xml ovisnost

```
<dependency>
  <groupId>com.fasterxml.jackson
    .dataformat</groupId>
  <artifactId>jackson-dataformat-xml
</artifactId>
</dependency>
```

Označena klasa modela

```
@XmlRootElement
public class Student {
  @XmlAttribute(name = "id")
  @XmlID
  private String id;
  @XmlElement private String ime;
}
```

Metoda kontrolera

```
@GetMapping(path = "/studenti",
  produces =
    APPLICATION_XML_VALUE)
public List<Student> dohvatiStudente() {
  return List.of(
    new Student("IT001", "Ana"),
    new Student("IT002", "Marko")
  );
}
```

XML izlaz

```
<?xml version="1.0" encoding="UTF-8"?>
<List>
  <item id="IT001"><n>Ana</n></item>
  <item id="IT002"><n>Marko</n></item>
</List>
```

✓ Prednosti: Standardni format, interoperabilnost, podrška starih sustava

⚠ Nedostaci: Opširniji od JSON-a; koristite JSON REST API-je gdje je moguće

03 — Tekstualni predlošci

Izlaz prostog teksta — nije potrebna nikakva oznaka. Idealno za e-poštu, izvještaje i konfigur. datoteke.

Konfiguracija Spring Boota

```
@Bean(name = "textTemplateEngine")
public TemplateEngine textEngine() {
    TemplateEngine e = new TemplateEngine();
    e.addTemplateResolver(textTemplateResolver());
    return e;
}

// Način: TemplateMode.TEXT
// Sufiks: .txt
```

text-predlozak.txt

```
Ime: [(${ime})]
URL: [(${url})]
Oznake:
[# th:each="o : ${oznake}" ]
    [(${o})]
[/]
```

Uporaba u Javi

```
Context ctx = new Context();
ctx.setVariable("ime", "Spring Framework");
ctx.setVariable("url", "https://spring.io");
ctx.setVariable("oznake", new String[]{ "#java", "#spring" });
String izlaz = textEngine.process("text-predlozak", ctx);
```

Izlaz

```
Ime: Spring Framework
URL: https://spring.io
Oznake:
    #java
    #spring
```

04 — JavaScript i CSS predlošci

Integracija statičkih resursa i inline Thymeleaf izrazi unutar JS/CSS datoteka

Statički CSS (src/main/resources/static/styles/main.css)

```
h2 { font-family: sans-serif;
      font-size: 1.5em; }
strong { font-weight: 700;
          background-color: yellow; }
p { font-family: sans-serif; }
```

HTML oznaka linka

```
<link th:href="@{/styles/main.css}"
      rel="stylesheet" />
```

actions.js + inline proslijeđivanje varijabli

```
function prikaziIme(ime) {
    alert("Pozdrav: " + ime);
}

// U Thymeleaf predlošku:
<script th:inline="javascript">
    var imeJs = /*[[${ime}]]*/;
</script>
<button th:onclick=
    "prikaziIme(imeJs);">Klikni</button>
```

u odnosu na alternative:

React/Vue

CSR — bez serverskih predložaka

JSP EL

Sličan, ali stariji pristup

FreeMark
er

Moćniji CSS predlošci

✓ Prednosti: @{} razrješavanje URL-ova, podrška inline izraza ⚠ Nedostaci: Promjene statičke datoteke zahtijevaju ponovnu primjenu; dinamika CSS-a ograničena

05 — Fragment predlošci (NOVO)

Komponente rasporeda koje se mogu ponovo koristiti — zaglavlje, podnožje, nav, modali — uključeni bilo gdje

Žičani okvir rasporeda

`th:fragment="zaglavlje" — Navigacijska traka`

Glavni sadržaj
(`th:replace="fragmenti/raspored :: sadrzaj"`)

Bočni
panel

`th:fragment="podnozje" — Podnožna traka`

Definicija i uporaba fragmenta

```
<!-- fragmenti/raspored.html -->
<nav th:fragment="zaglavlje">
  <a th:href="@{/}">Pocetna</a>
  <a th:href="@{/studenti}">Studenti</a>
</nav>
<footer th:fragment="podnozje">
  © 2026 Java Web Kolegij
</footer>
<!-- U predlošku stranice: -->
<div th:replace=
  "~{fragmenti/raspored :: zaglavlje}">
</div>

<!-- Parametrizirani fragment: -->
<div th:fragment="kartica(naslov,tijelo)">
  <h3 th:text="{naslov}"></h3>
  <p th:text="{tijelo}"></p>
</div>
```

06 — E-mail predlošci (NOVO)

Generirajte bogati HTML e-mail koristeći Thymeleaf + Spring Mail + JavaMailSender

Žičani okvir e-maila

Naziv vaše aplikacije

Dobrodošli, [(\${ime})]!

Hvala vam na registraciji. Vaš račun
uspješno je aktiviran.

Potvrdite e-mail →

© 2025 Java Web Kolegij

Primili ste ovu poruku jer ste se prijavili.

Java servis + predložak

```
// dobrodoslica-email.html u templates/email/
@Service
public class EmailServis {

    @Autowired JavaMailSender mailer;

    @Autowired SpringTemplateEngine tpl;

    public void posaljiDobrodoslicu(String na, String ime) {
        Context ctx = new Context();
        ctx.setVariable("ime", ime);
        String html = tpl.process( "email/dobrodoslica", ctx);
        MimeMessage msg = mailer.createMimeMessage();
        MimeMessageHelper h = new MimeMessageHelper(msg, true);
        h.setTo(na);
        h.setSubject("Dobrodosli!");
        h.setText(html, true);
        mailer.send(msg);
    }
}
```

07 — Sigurnost i i18n integracija (NOVO)

Thymeleaf Security dijalekt za vidljivost temeljenu na ulogama + Spring i18n za višejezična sučelja

Thymeleaf Security dijalekt

Ovisnost:

`thymeleaf-extras-springsecurity6`

```
<!-- Prikaži samo za ADMIN ulogu -->
<div sec:authorize=
    "hasRole('ADMIN')">
    <a href="/admin">Admin panel</a>
</div>

<!-- Prikaži prijavljeno korisničko ime -->
<span sec:authentication=
    "name">Korisnik</span>

<!-- Uvjetno na temelju autentifikacije -->
```

Prednosti sigurnosti:

Sučelje temeljeno na ulogama bez Java koda u HTML-u; besprijeekorna integracija sa Spring Security

Internacionalizacija (i18n)

Datoteke messages.properties:

`# messages_en.properties`

`greeting=Zdravo, {0}!`

`nav.students=Studenti`

`# messages_hr.properties`

`greeting=Dobar dan, {0}!`

`nav.students=Studenti`

`<!-- U Thymeleaf predlošku: -->`

`<h1 th:text=`

`"#{greeting(${korisnik.ime})}">`

`</h1>`

`<a th:text="#{nav.students}">`

Prednosti i18n:

Jednostavna `#{ključ}` sintaksa; lokalitet se automatski otkriva iz browser Accept-Language zaglavlja

08 — Neobrađeni (RAW) predlošci (NOVO)

RAW način: Thymeleaf obrađuje izraze, ali sadržaj izlazi s minimalnom transformacijom

Konfiguracija RAW načina

```
ClassLoaderTemplateResolver r =  
    new ClassLoaderTemplateResolver();  
r.setTemplateMode(TemplateMode.RAW);  
r.setPrefix("/templates/raw/");  
r.setSuffix(".raw");  
  
// Primjer RAW datoteke predloška:  
// (raw-izlaz.raw)  
Zdravo [{ime}]]!  
  
Ovo je neobrađeni izlaz.  
[({rawHtmlSadržaj})]
```

Usporedba: RAW vs TEXT vs HTML

Način	Escapiranje	Oznake	Najprikladnije za
HTML	Da	Da	Web stranice
TEXT	Da	Ne	E-mailovi, konfiguracija
RAW	Ne	Ne	Unaprijed izgrađeni sadržaj
XML	Da	Da	API odgovori
JS	JS	Ne	Dinamične skripte
CSS	CSS	Ne	Dinamični stilovi



Prednosti:

Nema HTML escapiranja — korisno za unaprijed prikazani HTML sadržaj, ugrađivanje JSON blokova ili izlaz blizak binarnom sadržaju



Nedostaci:

XSS rizik ako se korisnički podaci stave u RAW način — uvijek sanitizirajte unos! Rijetko potrebno u tipičnim web aplikacijama.



Primjeri: CMS gdje je sadržaj unaprijed sanitizirani HTML, JSON API odgovori, cjevovodi za markdown-u-HTML izlaz

Thymeleaf nasuprot alternativnim pogonima predložaka

Kada odabrati Thymeleaf — i kada razmotriti alternative

Thymeleaf

- ✓ Prirodni predlošci, Spring integracija, svi načini, Security dijalekt
- ⚠ Sporiji od Mustacha; nije idealan za čiste API-je
- ✚ Spring MVC web aplikacije, SSR

FreeMarker

- ✓ Vrlo moćni makroi, brz, zreo
- ⚠ Manje Spring-native, strmija krivulja učenja
- ✚ Složeno generiranje dokumenata

Mustache

- ✓ Bez logike, najbrži, višejezičan
- ⚠ Nema logike u predloščima (ponekad ograničavajuće)
- ✚ Jednostavno prikazivanje, više jezika

JSP/JSTL

- ✓ Standard Java EE, široko poznat
- ⚠ Miješa Javu u HTML, zastarjelo
- ✚ Stare poslovne aplikacije

Velocity

- ✓ Lagano, brzo prikazivanje
- ⚠ Neaktivan projekt, manja Spring podrška
- ✚ E-mail predlošci, stari sustavi

React/Vue

- ✓ Bogata interaktivnost, CSR/SSR hibrid
- ⚠ Nije serverski Java; treba REST API
- ✚ SPA, interaktivne nadzorne ploče

💡 Zaključak: Za Spring Boot web aplikacije, Thymeleaf je zadani izbor. Za zadatke samo s e-mailom, Mustache je lakši. Za bogatu interaktivnost, kombinirajte Thymeleaf rasporede s JS okvirima.

Reference i daljnje čitanje

Službena Thymeleaf dokumentacija

<https://www.thymeleaf.org/doc/tutorials/3.1/usingthymeleaf.html>

Thymeleaf rasporedi s fragmentima

<https://www.thymeleaf.org/doc/articles/layouts.html>

Spring Boot Thymeleaf integracija

<https://docs.spring.io/spring-boot/docs/current/reference/html/web.html>

Thymeleaf i18n i internacionalizacija

<https://www.baeldung.com/spring-boot-internationalization>

Thymeleaf Spring Security dodatak

<https://github.com/thymeleaf/thymeleaf-extras-springsecurity>

Manning: Spring Boot u praksi

<https://livebook.manning.com/book/spring-boot-in-practice/appendix-b/5>

Spring Mail + Thymeleaf e-mailovi

<https://www.baeldung.com/spring-email-thymeleaf>

Baeldung: Thymeleaf CSS i JS

<https://www.baeldung.com/spring-thymeleaf-css-js>

Hvala!

Java Web Programiranje — Thymeleaf Predlošci

Pitanja?