



**Blade: It will
most definitely
make the cut**

ZORAN ČAVUŽIĆ

11.10.2022.



- Jedna od Top 10 IT kompanija u Europi
- 27 000 zaposlenih
- Asseco SEE
- Mobile, Authentication, Voice

Zašto sam odabrao Blade framework?

- potreba za in-house utility aplikacijama
 - tehnologija koja nije u mojoj domeni
 - kratak rok za učenje
 - kratak rok za isporuku
 - unutar projekta ne postoji vrijeme rezervirano za razvoj tih aplikacija
 - ograničenja razvojnih okolina
- EnvMonitor
 - MyDBAdmin

Sadržaj

- Zašto sam odabrao Blade framework?
- Let's Blade Project
- Značajke
- Netty
- Arhitektura
- Primjeri
- Za/protiv
- Literatura

Let's Blade Project

- Po uzoru na Node Express, Python Flask, Macaron/Martini
- Cilj: da korisnik razumije cijeli framework u roku jednog dana
- Captcha, konverzija JSON-a, web predlošci, konekcija s bazom podataka, autentifikacija i autorizacija itd.
- Java 8+



Značajke

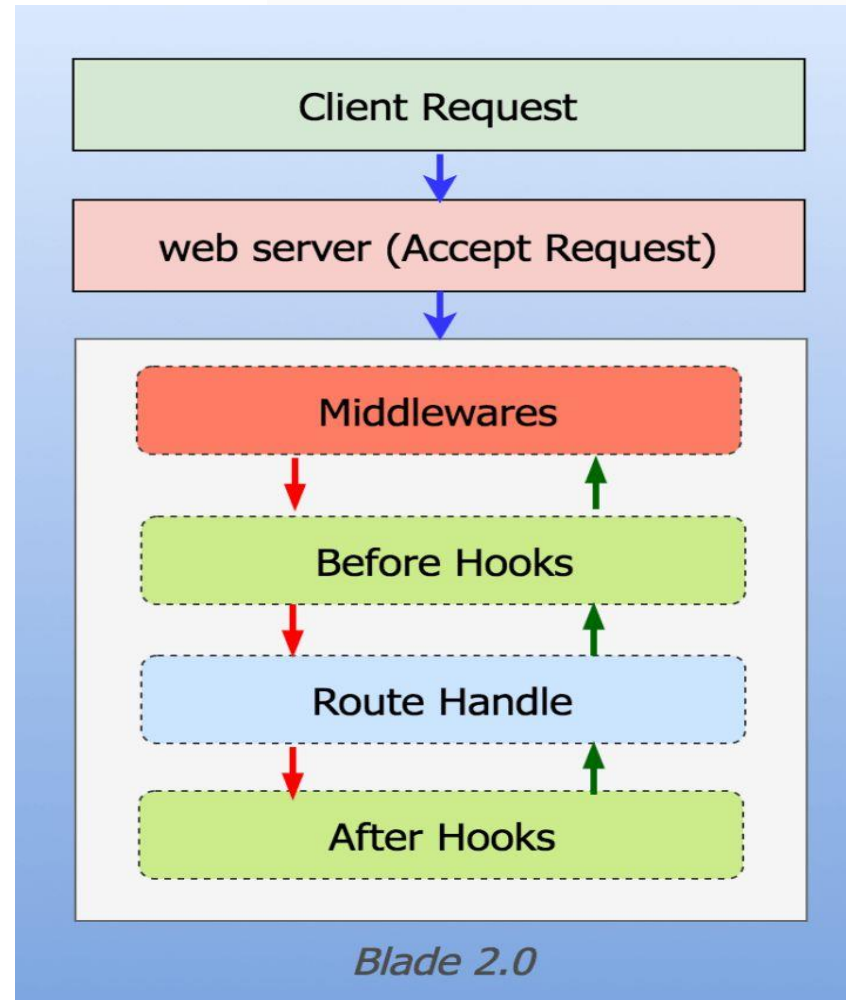
- Jednostavan
- Ravna krivulja učenja
- Samodostatan - nisu potrebni drugi dependency-ji
- Brz - 100 qps, 20 wps
- Podrška za webjars resurse
- Taskovi temeljeni na CRON izrazima
- XSS i CSRF zaštita
- RESTful API



Netty

- asinkroni event-driven network application framework
- može implementirati protokole poput FTP, SMTP, HTTP ili custom protokole
- non-blocking framework
- SSL/TLS podrška

Arhitektura



- Netty dobiva request
- Middleware
- WebHooks
- Usmjeravanje
- Response se šalje klijentu
- Cleanup

Primjeri

- **Kreirati novi Maven projekt**

```
<dependency> <groupId>com.bladejava</groupId>  
<artifactId>blade-mvc</artifactId>  
<version>2.0.14.RELEASE</version> </dependency>
```

- **Buildanje**

- JAR
- Maven Assembly Plugin technique

- **Pokretanje**

```
java -jar target/sample-blade-app.jar
```

- **Korištenje**

```
http://localhost:9000
```

Routing – usmjeravanje

- mehanizam za povezivanje URL-a i controllera
- može biti ‘basic’ (mikroservisi i jednostavne web aplikacije) i ‘anotated’

```
Blade.of()  
.get("/basic-routes-example", ctx -> ctx.text("GET called"))  
.post("/basic-routes-example", ctx -> ctx.text("POST called"))  
.put("/basic-routes-example", ctx -> ctx.text("PUT called"))  
.delete("/basic-routes-example", ctx -> ctx.text("DELETE called")) .start(App.class, args);
```

```
@Path public class RouteExampleController {  
  @GetRoute("/routes-example")  
  public String get(){ return "get.html"; }  
  @PostRoute("/routes-example")  
  public String post(){ return "post.html"; }  
  @PutRoute("/routes-example")  
  public String put(){ return "put.html"; }  
  @DeleteRoute("/routes-example")  
  public String delete(){ return "delete.html"; } }
```

```
@Route(value="/another-route-example",  
method=HttpMethod.GET)  
public String anotherGet()  
{ return "get.html" ; }
```

Parameter injection

Form parameter:

```
@GetRoute("/home")  
public void formParam(@Param String name){ System.out.println("name: " + name); }
```

Restful parameter:

```
@GetRoute("/users/:uid")  
public void restfulParam(@PathParam Integer uid){ System.out.println("uid: " + uid); }
```

File upload parameter:

```
@PostRoute("/upload")  
public void fileParam(@MultipartParam FileItem fileItem){byte[] file = fileItem.getData();  
}
```

Header parameter:

```
@GetRoute("/header")  
public void headerParam(@HeaderParam String referer){ System.out.println("Referer: " + referer); }
```

Parameter injection

Cookie parameter:

```
@GetRoute("/cookie") public void cookieParam(@CookieParam String myCookie){  
System.out.println("myCookie: " + myCookie); }
```

Body parameter:

```
@PostRoute("/bodyParam") public void bodyParam(@BodyParam User user){  
System.out.println("user: " + user.toString()); }
```

Value Object parameter, called by sending its attributes to the route:

```
@PostRoute("/voParam") public void voParam(@Param User user){ System.out.println("user: "  
+ user.toString()); }
```

```
<form method="post">  
  <input type="text" name="age"/>  
  <input type="text" name="name"/>  
</form>
```

Statički resursi

Blade također može posluživati statičke resurse tako da ih stavimo nr. u */resources/static* folder. Npr. *src/main/resources/static/app.css* će biti dostupan na *http://localhost:9000/static/app.css*.

```
blade.addStatics("/custom-static");  
blade.showFileList(true);
```

Statički resursi

Statički resursi zapakirani kao WebJar su također opcija.

```
<dependency> <groupId>org.webjars</groupId>  
<artifactId>bootstrap</artifactId>  
<version>4.2.1</version> </dependency>
```

U ovom slučaju resursi će biti dostupni na <http://localhost:9000/webjars/bootstrap/4.2.1/css/bootstrap.css>

Form parameters

Kod čitanja parametara s forme query metode vraćaju Optional objekt:

```
query(String name)  
queryInt(String name)  
queryLong(String name)  
queryDouble(String name)
```

Postoji i varijanta s defaultnom fallback vrijednosti:

```
String query(String name, String defaultValue)  
int queryInt(String name, int defaultValue)  
long queryLong(String name, long defaultValue)  
double queryDouble(String name, double defaultValue)
```

```
@PostRoute("/save") public void formParams(Request request){  
String username = request.query("username", „Zoran"); }
```

JSON podaci

```
curl -X POST http://localhost:9000/users -H 'Content-Type: application/json' \ -d  
'{"name":„Zoran", "site":„Zoran.com"}'
```

POJO:

```
public class User {  
    @Getter @Setter private String name;  
    @Getter @Setter private String site; }  
  
@PostRoute("/users")  
public void bodyParams(Request request){  
    String bodyString = request.bodyToString();  
}
```


Dohvaćanje request headera

```
String header1 = request.header("a-header");  
String header2 = request.header("a-safe-header", "with a default value");  
Map<String, String> allHeaders = request.headers();
```



Utility metode

Npr.

boolean isIE()

boolean isAjax()

String contentType()

String userAgent()

Output metode

Veoma lako možemo kreirati response zajedno sa HTTP code 200 i pripadajućim Content-Typeom.

Plain text:

```
response.text("Hello World!");
```

HTML:

```
response.html("<h1>Hello World!</h1>");
```

XML:

```
response.xml("<Msg>Hello World!</Msg>");
```

JSON:

```
response.json("{\"The Answer\":42}");
```

POJO sa automatskom konverzijom u JSON:

```
User user = new User(„Zoran“, „zoran.com“); response.json(user);
```

Download datoteke nikad nije bio lakši:

```
response.download("the-file.txt", "/path/to/the/file.txt");
```

Ostale operacije

- *Response status(int status)*
- *Map headers()*
- *Response notFound()*
- *Map cookies()*
- *Response contentType(String contentType)*
- *void body(@NonNull byte[] data)*
- *Response header(String name, String value)*
- *cookie(String name, String value)*
- *removeCookie(String name)*

WebHooks

Interceptor metode koje se izvršavaju prije i/ili poslije routing metode.

```
@FunctionalInterface public interface WebHook { boolean  
before(RouteContext ctx); default boolean  
after(RouteContext ctx) { return true; } }
```

Predložci

Template engine po želji možemo dodati kao Maven dependency:

```
<dependency> <groupId>com.bladejava</groupId>  
<artifactId>blade-template-jetbrick</artifactId>  
<version>0.1.3</version> </dependency>
```

```
@Bean  
public class TemplateConfig implements BladeLoader {  
    @Override  
    public void load(Blade blade) { blade.templateEngine(new  
        JetbrickTemplateEngine()); } }
```

Logiranje

Blade koristi već integrirani slf4j api sa blade-log konfiguracijom.

Može se koristiti i bilo koji drugi logger, ali je potrebno isključiti defaultni.

```
<dependency>
  <groupId>com.bladejava</groupId>
  <artifactId>blade-mvc</artifactId>
  <version>${blade.version}</version>
  <exclusions>
    <exclusion>
      <groupId>com.bladejava</groupId>
      <artifactId>blade-log</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

Za/Protiv

■ ZA

- Jednostavnost
- Brzo učenje
- Lib ima manje od 500 kB
- Mali utrošak resursa
- Netty

■ PROTIV

- Nedostatak literature
- Nedostatak primjera
- Slaba/nepostojeća podrška
- Manje detaljan i konfigurabilan od npr. Springa
- Manje pogodan za enterprise aplikacije

Literatura

- <https://github.com/lets-blade/blade>
- <https://www.baeldung.com/blade>
- <https://www.baeldung.com/netty>
- <https://www.knowledgefactory.net/2020/04/blade-java-web-framework-guide.html>

Hvala!

PITANJA?