

Tudo o que você queria saber sobre MVC no Jakarta EE e tinha vergonha de perguntar

Daniel Dias
Jakarta EE Developer -

SouJava

@danieldiasjava



Agenda

Jakarta MVC

Demo

Q&A

Jakarta MVC

Jakarta MVC

- **Framework orientada a ação construída sobre as camadas da JAX-RS.**

Jakarta MVC

- **Framework orientada a ação construída sobre as camadas da JAX-RS.**
- **Aproveitar as tecnologias Java EE existentes.**

Jakarta MVC

- **Framework orientada a ação construída sobre as camadas da JAX-RS.**
- **Aproveitar as tecnologias Java EE existentes.**
- **Integração com CDI e Bean Validation**

Jakarta MVC

- **Framework orientada a ação construída sobre as camadas da JAX-RS.**
- **Aproveitar as tecnologias Java EE existentes.**
- **Integração com CDI e Bean Validation**
- **Fornece suporte interno para JSP e Facelets.**

Jakarta MVC

- **Framework orientada a ação construída sobre as camadas da JAX-RS.**
- **Aproveitar as tecnologias Java EE existentes.**
- **Integração com CDI e Bean Validation**
- **Fornece suporte interno para JSP e Facelets.**
- **Contém suporte RestEasy, Jersey e Apache CXF através de módulos.**

Jakarta MVC

- **Framework orientada a ação construída sobre as camadas da JAX-RS.**
- **Aproveitar as tecnologias Java EE existentes.**
- **Integração com CDI e Bean Validation**
- **Fornece suporte interno para JSP e Facelets.**
- **Contém suporte RestEasy, Jersey e Apache CXF através de módulos.**
- **Não se destina a substituição do frameworks baseado em componentes como JSF, mas simplesmente oferecer uma alternativa para construção de aplicações web no Java EE.**

Controller

Controller

```
public class HelloController {
```


13 / 32

View

View

```
@Controller
@Path("hello")
public class HelloController {

    @GET
    public String getHello() {
        return "hello.jsp";
    }
}
```

View

```
@Controller
@Path("hello")
public class HelloController {

    @GET
    public Response getHello() {
        return Response.status(OK)
            .entity("hello.jsp")
            .build();
    }
}
```


View

```
@Controller
@Path("hello")
public class HelloController {

    @View("hello.jsp")
    @GET
    public void getHello() {

    }
}
```

- `javax.mvc.View`

View

```
@Controller
@Path("hello")
public class HelloController {

    @GET
    public Viewable getHello() {
        return new Viewable("hello.jsp");
    }
}
```

- Somente na RI - `org.eclipse.krazo.engine.Viewable`

View Engines suportadas na Krazo(RI)

AsciiDoc	.adoc .asciidoc
Freemarker	.ftl
Handlebars	.hbs .handlebars
Jade	.jade
Jetbrick	.jetx
JSR-223	.groovy .js .py
Mustache	.mustache
Pebble	.peb
Stringtemplate	.st
Thymeleaf	.html
Velocity	.vm
Jtwig	.twig.html .twig

Model

Model

```
@Controller  
@Path("hello")  
public class HelloController {
```

```
    @Inject  
    private Models model
```

```
    @View("hello.jsp")  
    @GET  
    public void getHello() {  
        model.put("MSG", "SouJava");  
    }
```

```
}
```

- `javax.mvc.Models`

Model - CDI

```
@Named("jug")  
@RequestScoped  
public class JUG {  
  
    private String nome;  
}
```

- javax.inject.Named;
- javax.enterprise.context.RequestScoped;

Model - CDI

```
@Controller
@Path("hello")
public class HelloController {

    @Inject
    private JUG jug;

    @View("hello.jsp")
    @GET
    public void getHello() {
        jug.setNome("SouJava");
    }
}
```

Model

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <title>MVC 1.0 Hello Demo</title>
  </head>
  <body>
    <h1>Hello ${MSG} usando Model</h1>
    <h1>Hello ${jug.nome} usando CDI</h1>
  </body>
</html>
```

- Suporte apenas JSP, Facelets e Thymeleaf .

BOOT

Boot

```
@ApplicationPath("resource")
public class MVCApplication extends Application {

    @Override
    public Map<String, Object> getProperties() {
        Map<String, Object> defaultFileExtension = new HashMap<>();
        defaultFileExtension.put(Properties.DEFAULT_VIEW_FILE_EXTENSION, "jsp");
        return defaultFileExtension;
    }
}
```

- **javax.ws.rs.ApplicationPath**
- **javax.ws.rs.core.Application**



JAKARTA EE

"Adopt a Spec" Program for JUGs

<https://jakarta.ee/community/adopt-a-spec/>

Adopt-a-Spec

- Código:
 - (<https://github.com/eclipse-ee4j/mvc-api>)
 - (<https://github.com/eclipse-ee4j/krazo>)
 - (<https://github.com/eclipse-ee4j/mvc-tck>)
- Issues :
 - (<https://github.com/eclipse-ee4j/mvc-api/issues>)
 - (<https://github.com/eclipse-ee4j/krazo/issues>)
- Site (<https://www.mvc-spec.org/>)
- Eclipse Proposals:
 - (<https://projects.eclipse.org/proposals/eclipse-krazo>)
 - (<https://projects.eclipse.org/proposals/jakarta-mvc>)
- Lista de E-mail :
 - (mvc-dev@eclipse.org)
 - (krazo-dev@eclipse.org)

DEMO



<https://git.io/fjFbu>

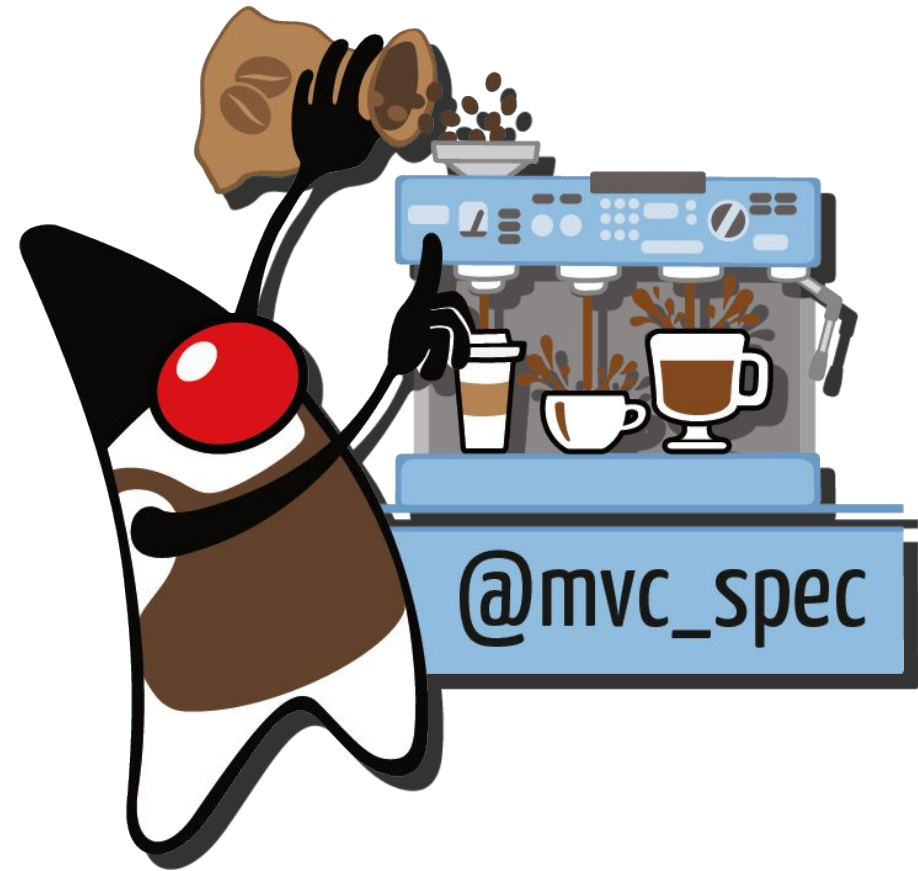
Referências

- <https://projects.eclipse.org/projects/ee4j.krazo>
- <https://www.mvc-spec.org/>
- <https://jcp.org/en/jsr/detail?id=371>
- <https://medium.com/danieldiasjava/jsr-371-mvc-1-0-com-tomcat-tomee-45f9638f276c>
- <https://medium.com/danieldiasjava/jsr-371-mvc-1-0-validação-248f58e370dc>
- <https://medium.com/danieldiasjava/criando-uma-aplicação-com-jsr-371-mvc-1-0-13635d0fc41f>
- <http://blog.triadworks.com.br/mvc-1-0-jsr-para-um-framework-mvc-action-based-na-java-ee-8>
- <http://blog.caelum.com.br/primeiros-passos-do-mvc-1-0/>

Q&A



OverOps



Obrigado !

Daniel Dias

Jakarta EE Developer -

SouJava

@danieldiasjava