JAVA EE COURSE

JAVASERVER FACES AND JAVA EE

By the expert: Eng. Ubaldo Acosta

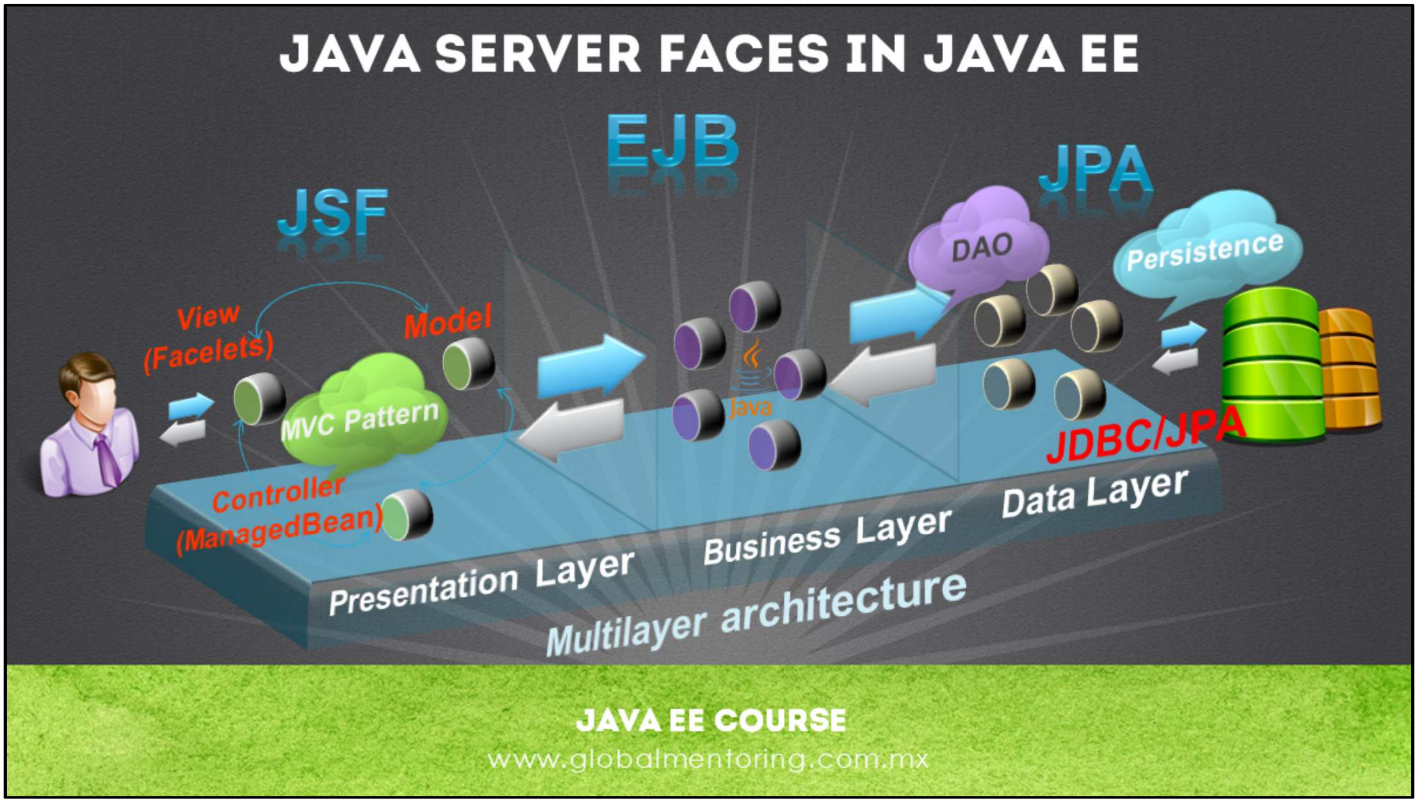
JAVA EE COURSE

www.globalmentoring.com.mx

Hello, Ubaldo Acosta greets you again. I hope you're ready to start with this lesson ..

We are going to study the role played by JavaServer Faces technology in Java EE.

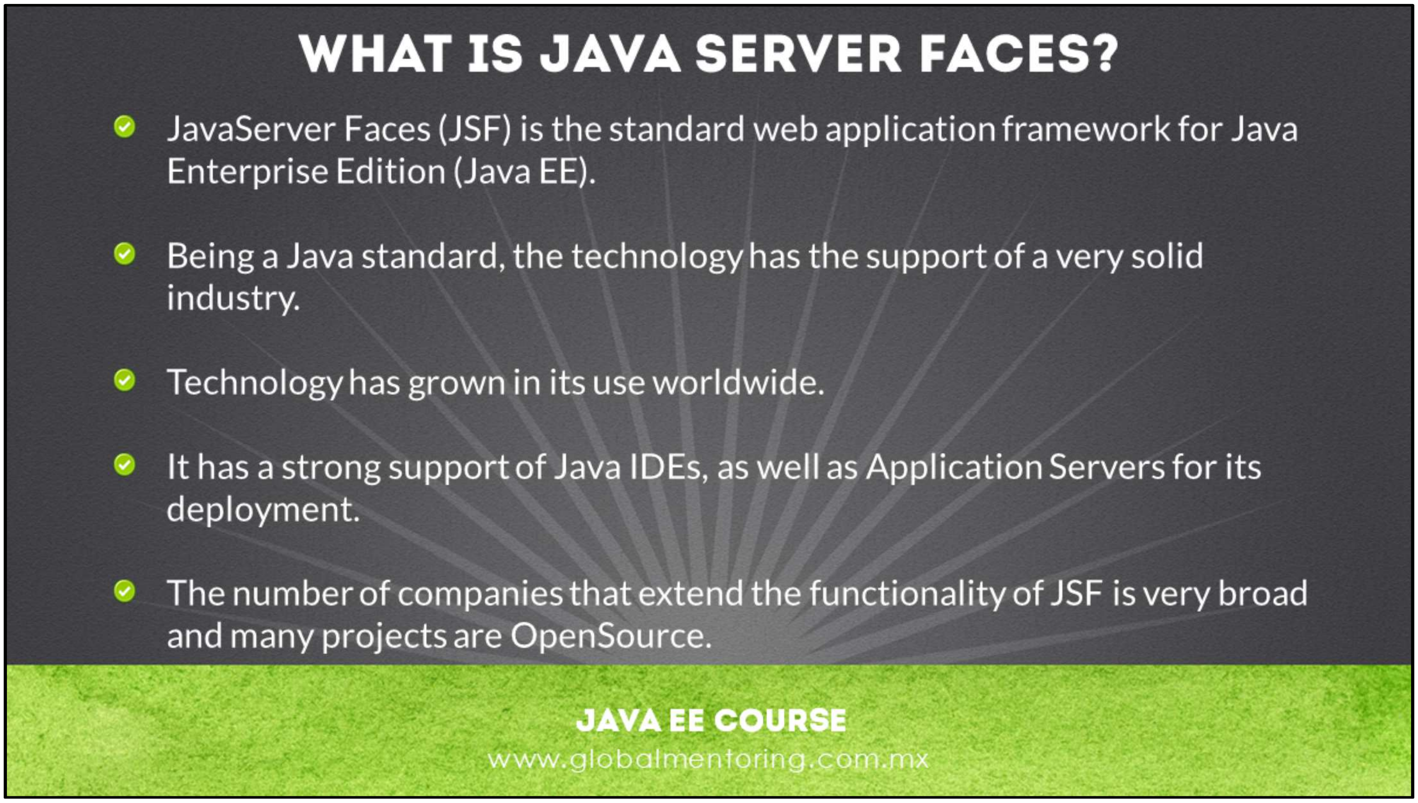
Are you ready? Let's go!



In the figure we can see the role of JSF technology in a Java Enterprise architecture. JSF applies directly to the presentation layer, which is responsible for tasks such as:

- Generation of Presentation Code, with technologies such as HTML, CSS and JavaScript with previously created or customized components.
- Processing of HTTP requests from the client through HTML forms, with the possibility of adding AJAX requests.
- Validation of parameters received in the HTTP request.
- Converters to work directly with types such as Date, Boolean, Integer, etc.
- Recovery of model information, relying on the Service Layer (EJB's) for this.
- Processing of the response and selection of the view to be shown to the client, applying the concept of Navigation. Among several other tasks.

In this lesson we will review how to integrate JSF, EJB and JPA technologies into a three-tier Java EE architecture. To study the JSF technology in more detail, we recommend our JavaServer Faces course.



WHAT IS JAVA SERVER FACES?

- ✓ JavaServer Faces (JSF) is the standard web application framework for Java Enterprise Edition (Java EE).
- ✓ Being a Java standard, the technology has the support of a very solid industry.
- ✓ Technology has grown in its use worldwide.
- ✓ It has a strong support of Java IDEs, as well as Application Servers for its deployment.
- ✓ The number of companies that extend the functionality of JSF is very broad and many projects are OpenSource.

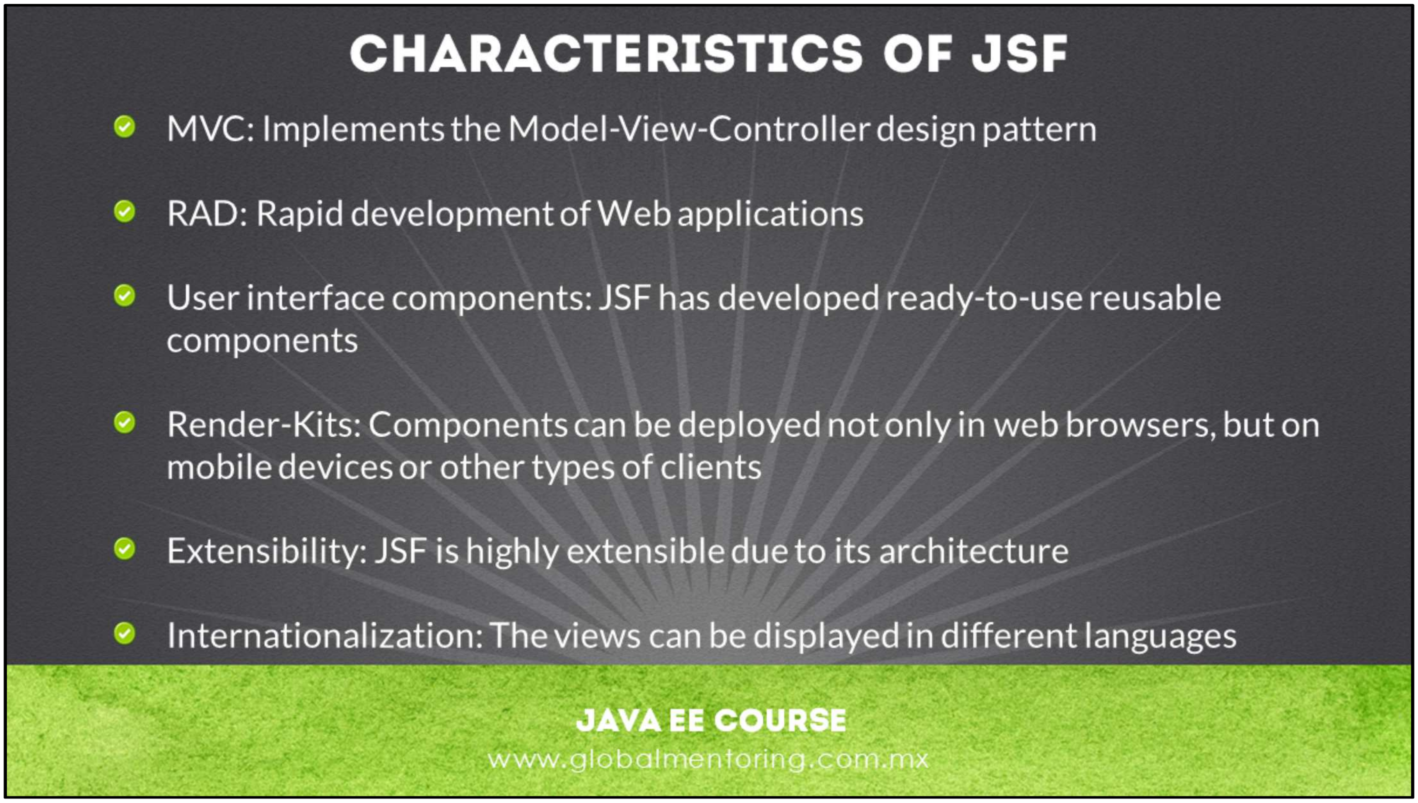
JAVA EE COURSE
www.globalmentoring.com.mx

JavaServer Faces (JSF) was designed to simplify the construction of user interface for Web applications. One of the key points was the support of the "tools" created for this type of Java applications.

JSF has the support of Java IDEs such as:
- Eclipse
- MyEclipse
- NetBeans
- IntelliJ IDEA
- BEA Workshop
- Oracle JDeveloper

It has strong support from Java Application Servers in order to support the applications created in JSF.

In addition, the number of companies that extend the functionality of JSF are several, and most of the projects are OpenSource. As the most interesting examples we have PrimeFaces, iceFaces, richFaces and openFaces.



CHARACTERISTICS OF JSF

- MVC: Implements the Model-View-Controller design pattern
- RAD: Rapid development of Web applications
- User interface components: JSF has developed ready-to-use reusable components
- Render-Kits: Components can be deployed not only in web browsers, but on mobile devices or other types of clients
- Extensibility: JSF is highly extensible due to its architecture
- Internationalization: The views can be displayed in different languages

JAVA EE COURSE
www.globalmentoring.com.mx

Some of the most important characteristics of JSF are:

- MVC: The JSF framework implements the Model-View-Controller design pattern, providing an event-oriented approach.

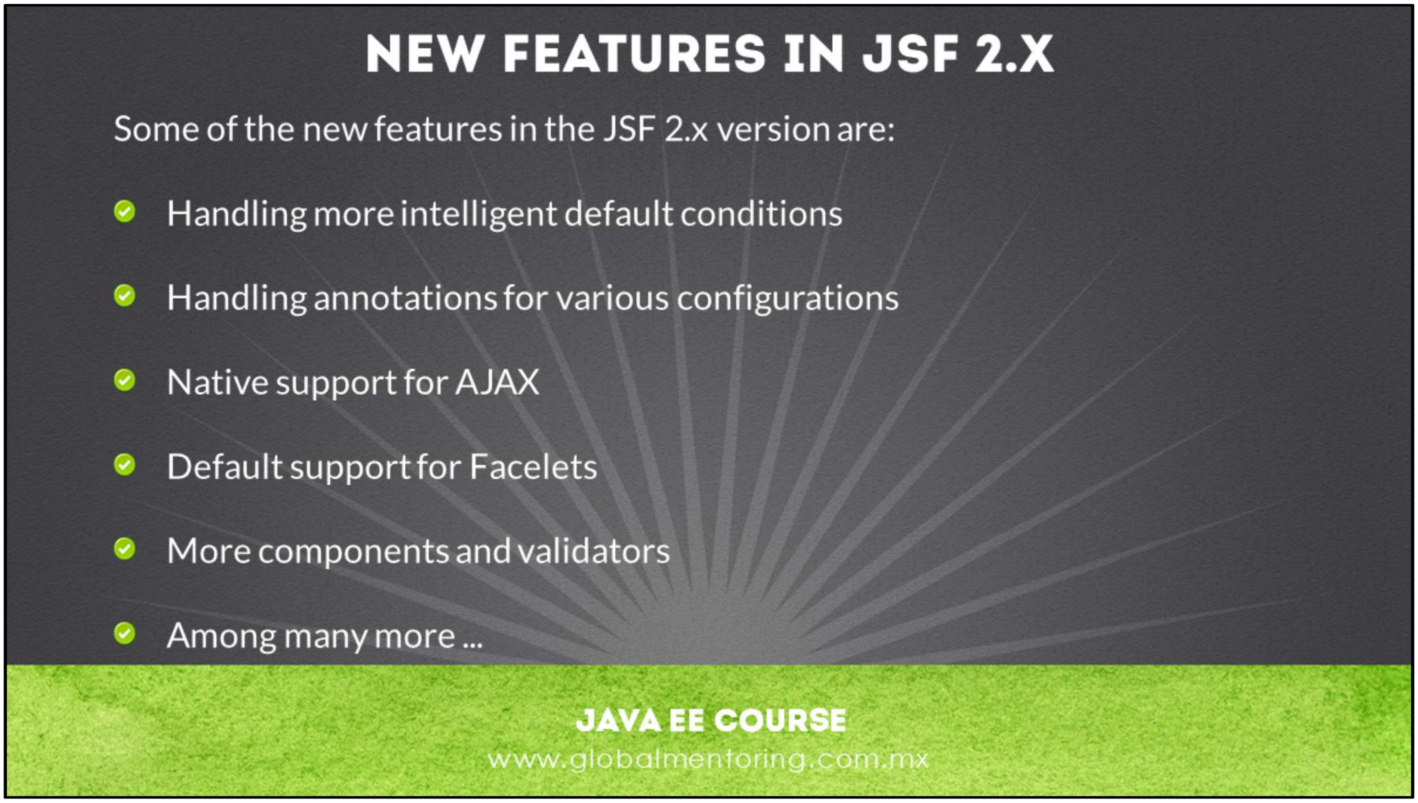
- RAD: Due to the number of ready-to-use components, JSF streamlines the development of Web applications for Java.

- User interface components (GUI): JSF provides an API to create complex HTML components, including JavaScript and integrated CSS, creating reusable components.

- Render-Kits: Components can be deployed not only in Web browsers, but in mobile devices or other types of clients.

- Extensibility: JSF allows to create new components more easily, so there are several frameworks that extend the power of JSF and Ajax, such as richFaces, iceFaces, among others.

- Internationalization: The views can be displayed in different languages.



NEW FEATURES IN JSF 2.X

Some of the new features in the JSF 2.x version are:

- ✓ Handling more intelligent default conditions
- ✓ Handling annotations for various configurations
- ✓ Native support for AJAX
- ✓ Default support for Facelets
- ✓ More components and validators
- ✓ Among many more ...

JAVA EE COURSE

www.globalmentoring.com.mx

The JSF 2.x version is a very important improvement for this Java technology. Although a new MVC standard is being proposed in Java, JSF is still the most used technology today. Some of the improvements are:

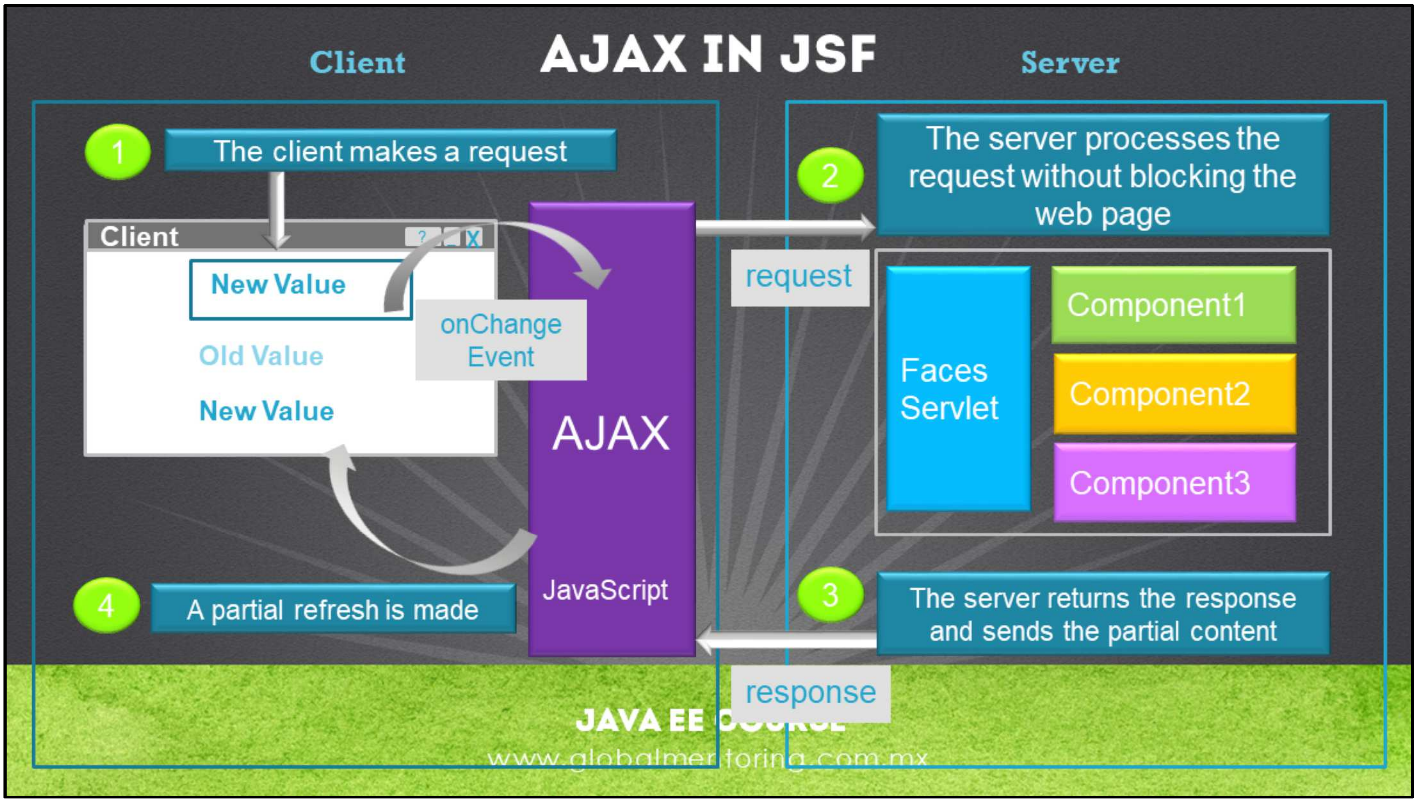
- Handling more intelligent default conditions: This applies to simple navigation cases, which are no longer required to be added to the configuration file faces-config.xml

- Handling annotations for various configurations: This greatly simplifies adding a Managed Bean to our application, avoiding its declaration in the faces-config.xml file. Later we will see the concept of Managed Beans.

- Native support for AJAX: AJAX technology is already part of the life cycle of JSF.

- Default support for Facelets: Facelets technology takes into account the life cycle of JSF, unlike JSP's.

- More components and validators: New components have been created and added to the standard JSF library. Among many more features.



AJAX IN JSF

Client

Server

1 The client makes a request

Client

New Value

Old Value

New Value

onChange Event

AJAX

JavaScript

request

2 The server processes the request without blocking the web page

Faces Servlet

Component1

Component2

Component3

3 The server returns the response and sends the partial content

response

4 A partial refresh is made

JAVA EE COURSE

www.globalmentoring.com.mx

AJAX is not a technology, but a set of technologies that allow us to improve the usability of our Web applications. AJAX is the acronym for Asynchronous JavaScript and XML.

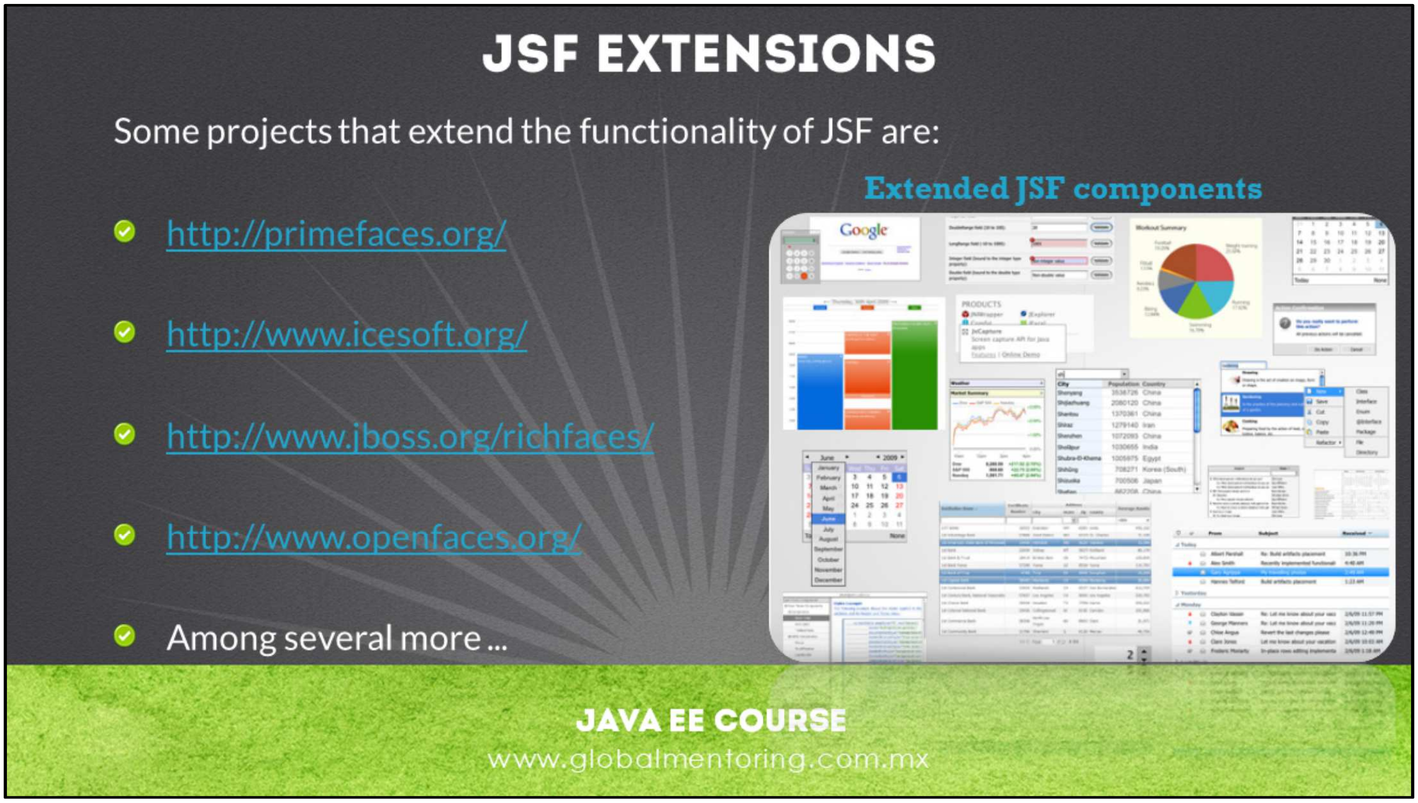
A classic Web application uses a client server interaction synchronously, this means that before each user action, it is necessary to wait for the Web server to finish processing said request and until then we can observe the HTML response by the server.

AJAX allows us to improve this behavior, adding the concept of asynchronous requests. This means that before a request from the user, it is not necessary to wait for the server to finish processing the request to continue working. On the other hand, the server receives the request, processes it and at the end returns a response to the client, however the Web page never stopped during this process and the client was able to perform other tasks.

A common example of AJAX processing is when creating an email in gmail. On the one hand we can attach a file (asynchronous request) and on the other hand we can continue writing our email. When the process of attaching the file is completed, we will be able to complete the sending of the email. This is just one example of all the improvements we can add to our Web interfaces by applying the concept of asynchronous requests with AJAX.

Another advantage of using AJAX is that we obtain partial refresh, because we can update only the elements of the Web page that we are interested in.

With JSF 2.x it is possible to add AJAX natively to our Web applications. In the JSF, EJB and JPA integration exercise we will add the use of AJAX to generate the people capture form.



JSF EXTENSIONS

Some projects that extend the functionality of JSF are:

- http://primefaces.org/
- http://www.icesoft.org/
- http://www.jboss.org/richfaces/
- http://www.openfaces.org/
- Among several more ...

Extended JSF components

JAVA EE COURSE

www.globalmentoring.com.mx

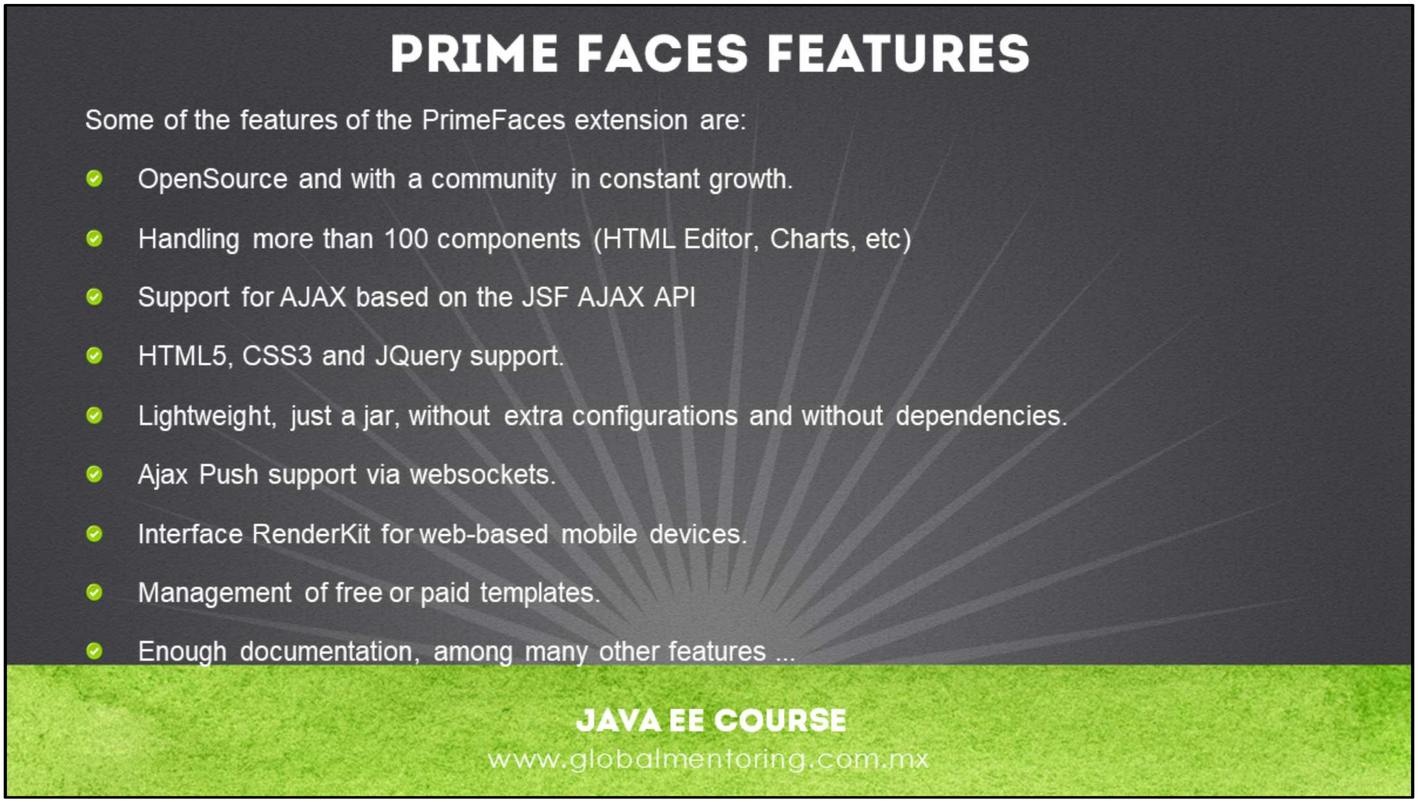
The number of components and functionality that comes by default in JSF is limited. However, creating new components and adding new functionality is one of the successes of this technology.

Therefore, many companies have created projects to extend the functionality and add new components, in addition to those already available by default.

The most popular extensions to date to add new components to our JSF pages are:

- http://primefaces.org/
- http://www.icesoft.org/
- http://www.jboss.org/richfaces/
- http://www.openfaces.org/
- Among several more…

In the project of integration of business technologies we will use the extension of PrimeFaces, for several reasons that we will comment later.



PRIME FACES FEATURES

Some of the features of the PrimeFaces extension are:

- OpenSource and with a community in constant growth.
- Handling more than 100 components (HTML Editor, Charts, etc)
- Support for AJAX based on the JSF AJAX API
- HTML5, CSS3 and JQuery support.
- Lightweight, just a jar, without extra configurations and without dependencies.
- Ajax Push support via websockets.
- Interface RenderKit for web-based mobile devices.
- Management of free or paid templates.
- Enough documentation, among many other features ...

JAVA EE COURSE
www.globalmentoring.com.mx

In this lesson, we will use the PrimeFaces extension to add several improvements to our Web interfaces when using JSF.

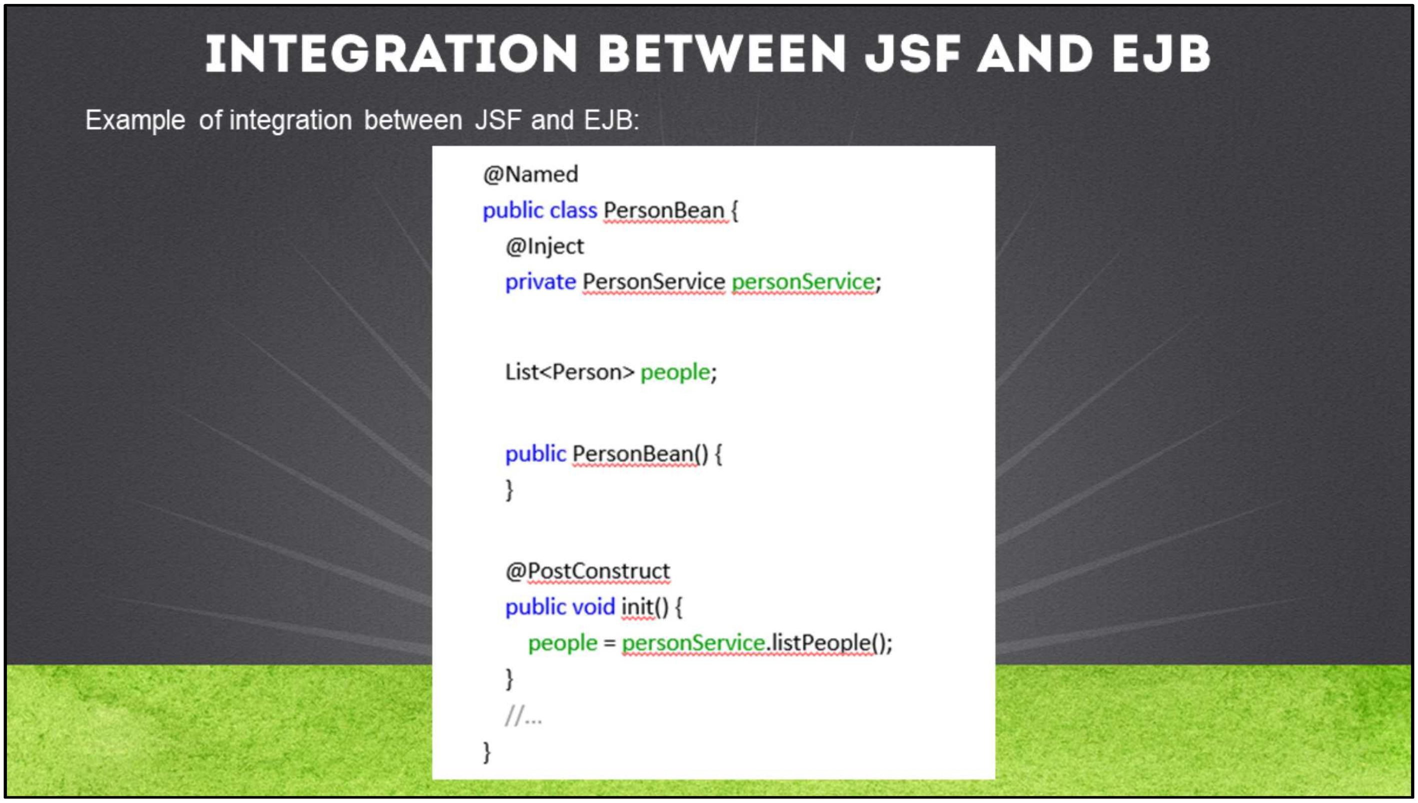
PrimeFaces is one of the most popular extensions to date, and has an excellent performance, allowing you to configure and use the new components in a very simple way.

To configure PrimeFaces, simply download the library and add it to the classpath of our project.

And to start using the components, we just add the namespace to our xhtml page, as follows:

```
<html xmlns="http://www.w3c.org/1999/xhtml"
     xmlns:h="http://java.sun.com/jsf/html"
     xmlns:p="http://primefaces.org/ui">
    …
   <h:body>
      <p:editor />
   </h:body>
</html>
```

In the technology integration exercise, we will use PrimeFaces to improve the look & feel and usability of our application, integrating AJAX and a CSS theme.



INTEGRATION BETWEEN JSF AND EJB

Example of integration between JSF and EJB:

```
@Named
public class PersonBean {
  @Inject
  private PersonService personService;


  List<Person> people;


  public PersonBean() {
  }


  @PostConstruct
  public void init() {
    people = personService.listPeople();
  }
  //...
}
```

As of today, integration between different business technologies is simpler each time.

As we can see in the following code, integrating an EJB to be used in a JSF ManagedBean is as simple as using the @EJB annotation and specifying the type of EJB to be used.

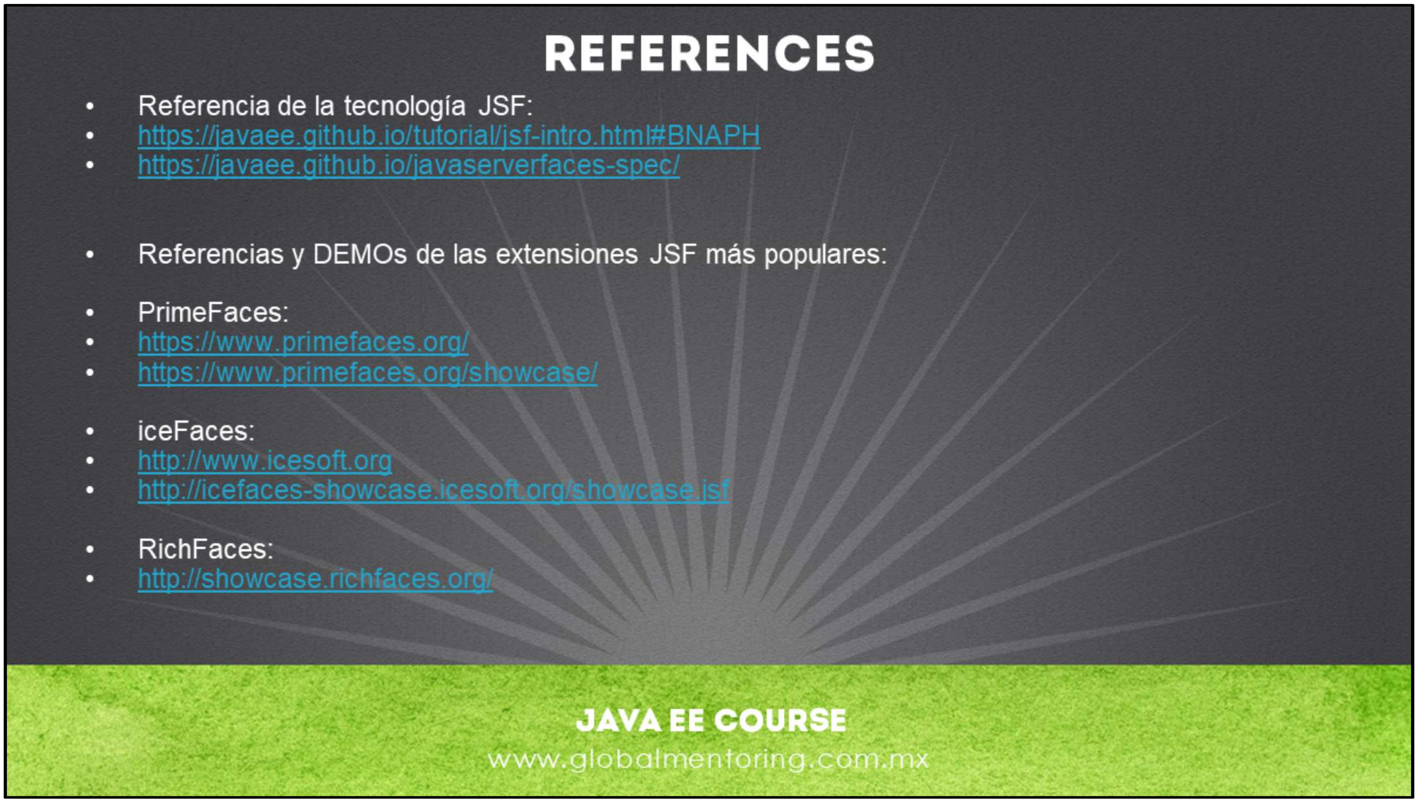
This will cause the application server to automatically search for an instance of the specified EJB type, and once localized, this dependency is automatically injected by the Java application server.

```
@Named
public class PersonBean {
  @Inject
  private PersonService personService;


  List<Person> people;


  public PersonBean() {
  }


  @PostConstruct
  public void init() {
    people = personService.listPeople();
  }
  //...
}
```



REFERENCES

- Referencia de la tecnología JSF:
- https://javaee.github.io/tutorial/jsf-intro.html#BNAPH
- https://javaee.github.io/javaserverfaces-spec/

- Referencias y DEMOs de las extensiones JSF más populares:

- PrimeFaces:
- https://www.primefaces.org/
- https://www.primefaces.org/showcase/

- iceFaces:
- http://www.icesoft.org
- http://icefaces-showcase.icesoft.org/showcase.jsf

- RichFaces:
- http://showcase.richfaces.org/

JAVA EE COURSE
www.globalmentoring.com.mx

We leave you the following references of JSF:

- https://javaee.github.io/tutorial/jsf-intro.html#BNAPH
- https://javaee.github.io/javaserverfaces-spec/

- References and DEMOs of the most popular JSF extensions:

- PrimeFaces:
- https://www.primefaces.org/
- https://www.primefaces.org/showcase/

- iceFaces:
- http://www.icesoft.org
- http://icefaces-showcase.icesoft.org/showcase.jsf

- RichFaces:
- http://showcase.richfaces.org/

ONLINE COURSE

JAVA EE
JAKARTA EE

By: Eng. Ubaldo Acosta

JAVA EE COURSE

www.globalmentoring.com.mx