

OBJECTIVE OF THE PROPOSAL

GENERAL OBJECTIVE

The main objective of the project is to upgrade the Weblogic application into Spring Boot based JSF application.

SPECIFIC OBJECTIVE

The following are specific objectives of the project attributing towards the achievement of the broad objective stated above.

- ❖ To study/research the conversion from legacy JSF to Spring Based JSF and gather the required information to implement the POC.
- ❖ To Define the purpose of the system in terms of functional and non-functional requirements.
- ❖ To Translate the legacy JSF files into upgraded JSF view which is compatible with spring application. This includes testing the compatibility of the tags used in the JSP pages and find suitable replacements wherever an issue arises.
- ❖ To perform a high-level regression of the system post migration.

METHODOLOGY

We are proposing 2 different solutions to upgrade weblogic-based application into spring based jsp application.

Solution-1 - Spring Boot Starter based implementation

Spring Boot based join-richfaces starter implementation, which provides the default configuration of spring boot and JSF environment, auto configuration of JSF related run time environment and spring boot related auto configuration with embedded Tomcat server. There is no need to add any additional web application configuration or any JSF configuration.

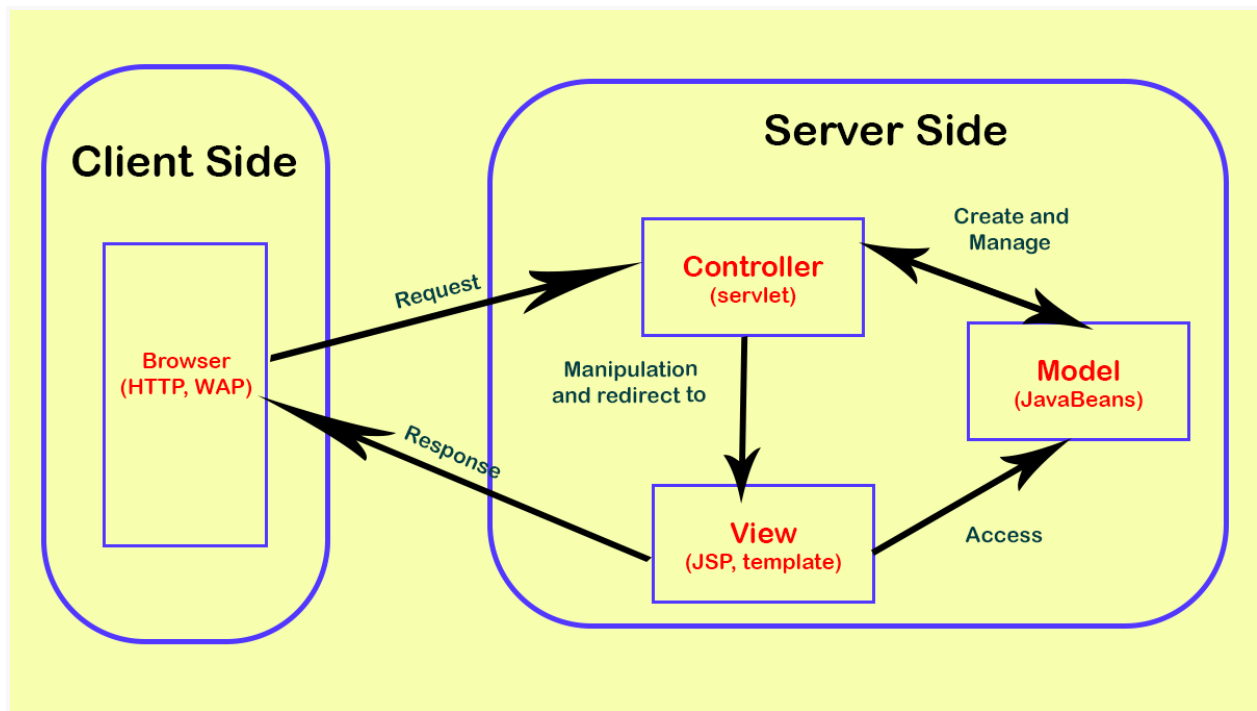
Pros

- ❖ Generate the base application from spring official portal and auto configuring of the application by default.
- ❖ It supports the spring boot version up to 2.7.5 with JSF 4.5.17.
- ❖ There is no more jsf configuration required.
- ❖ It provides the overriding option on top of the auto configuration.

- ❖ It is also providing starter package for my richface and auto configure the jsf related setting and environment properties.
- ❖ We can include addition starter package as per requirement.

Cons

- ❖ It does not support JSP files, only support JSF/XHTML files.
- ❖ We should manually migrate all JSP files into JSF/XHTML files and also convert all JSP related tags into the corresponding JSF tags.



Solution-2 - Customized JSF with Spring Boot implementation

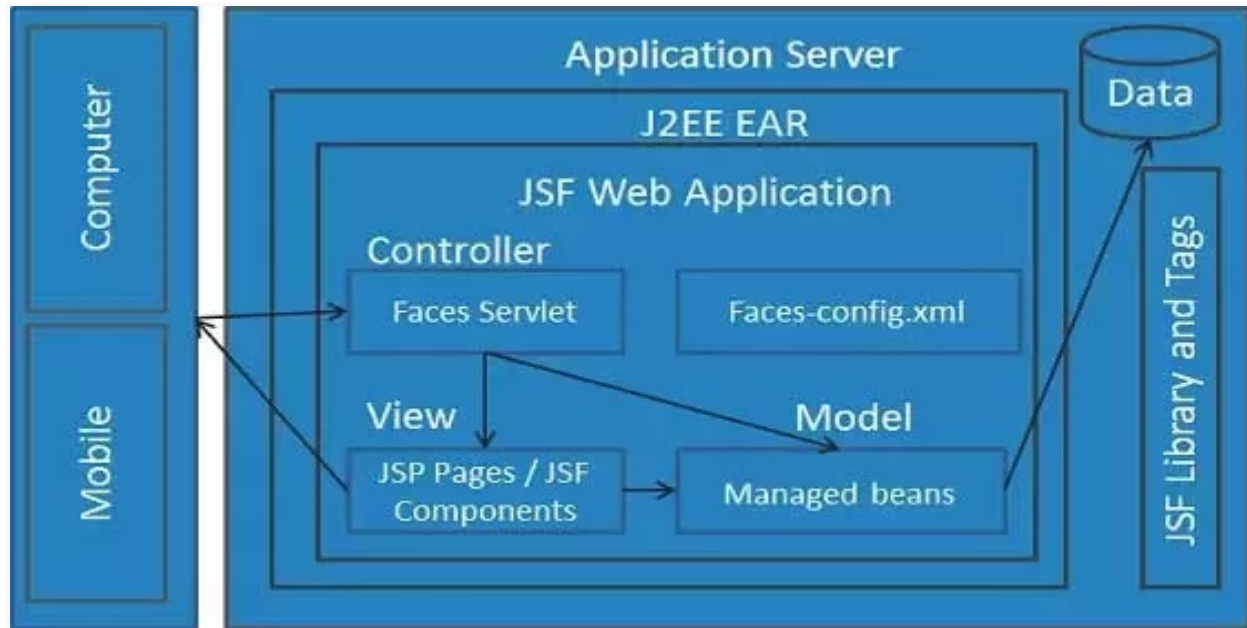
The customized Spring based JSF implementation, which provides the option to select the dependency and required or latest version. The developer is responsible for to configure the JSF environment, JSF configuration, Spring Boot configuration and etc.

Pros

- ❖ It provides the recent version of spring boot application.
- ❖ It provides option to include the available jsf dependency of richfaces/myfaces.
- ❖ We can add addition starter package as per requirement.

Cons

- ❖ We should manually configure JSF and Spring Boot configuration. It takes lot time to work the server up and running.
- ❖ It takes more time to research and test each page of JSF/JSP application and to confirm whether it will compatible with upgraded version of the application.
- ❖ If we use JSP, the system throwing error.



Recommendation

The solution 1 is recommended due to ease of implementation and more predictability in the migration approach. But, still we require more time to migrate each JSP into JSF related XHTML files. (tag related migration)