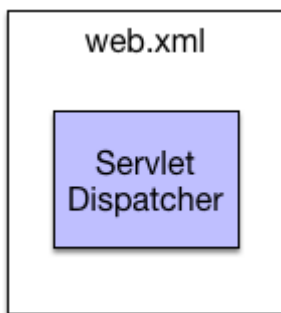


## CURSO SPRING FRAMEWORK GRATIS APUNTATE!!

El concepto de Spring Web Initializer es poco conocido. En muchas ocasiones cuando damos de alta una aplicación con Spring MVC necesitamos si o sí dar de alta un despachador a nivel del web.xml.



spring web initalizer

Sin embargo a partir de la versión 4 de Spring podemos inicializar Spring MVC sin hacer uso de web.xml. Vamos a ver como hacerlo. El primer paso es configurar las dependencias de Maven.

```
&lt;project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd"&gt;
    &lt;modelVersion&gt;4.0.0&lt;/modelVersion&gt;
    &lt;groupId&gt;com.arquitecturajava&lt;/groupId&gt;
    &lt;artifactId&gt;springweb&lt;/artifactId&gt;
```

## Spring Web Initializer ,eliminando el web.xml

```
&lt;version&gt;0.0.1-SNAPSHOT&lt;/version&gt;
&lt;packaging&gt;war&lt;/packaging
&lt;dependencies&gt;

    &lt;!- -
https://mvnrepository.com/artifact/org.springframework/spring-core - -
&gt;
    &lt;dependency&gt;
&lt;groupId&gt;org.springframework&lt;/groupId&gt;
    &lt;artifactId&gt;spring-
core&lt;/artifactId&gt;
&lt;version&gt;4.3.13.RELEASE&lt;/version&gt;
    &lt;/dependency&gt;
    &lt;!- -
https://mvnrepository.com/artifact/org.springframework/spring-context
--&gt;
    &lt;dependency&gt;
&lt;groupId&gt;org.springframework&lt;/groupId&gt;
    &lt;artifactId&gt;spring-
context&lt;/artifactId&gt;
&lt;version&gt;4.3.13.RELEASE&lt;/version&gt;
    &lt;/dependency&gt;

    &lt;!- -
https://mvnrepository.com/artifact/org.springframework/spring-webmvc -
-&gt;
    &lt;dependency&gt;
&lt;groupId&gt;org.springframework&lt;/groupId&gt;
    &lt;artifactId&gt;spring-
webmvc&lt;/artifactId&gt;
&lt;version&gt;4.3.13.RELEASE&lt;/version&gt;
```

```
        &lt;/dependency&gt;

        &lt;dependency&gt;
&lt;groupId&gt;javax.servlet&lt;/groupId&gt;
        &lt;artifactId&gt;javax.servlet-
api&lt;/artifactId&gt;
&lt;version&gt;3.1.0&lt;/version&gt;
&lt;scope&gt;provided&lt;/scope&gt;
        &lt;/dependency&gt;
&lt;/dependencies&gt;

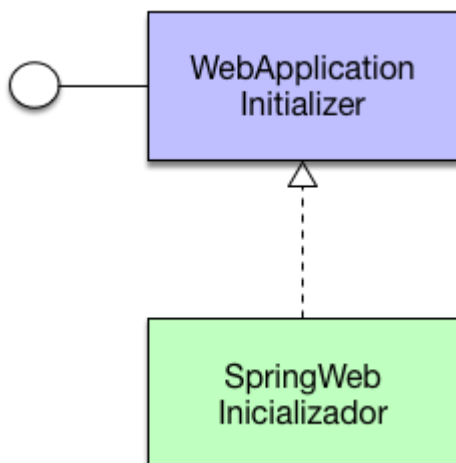
&lt;build&gt;
    &lt;plugins&gt;
        &lt;plugin&gt;
            &lt;artifactId&gt;maven-
compiler-plugin&lt;/artifactId&gt;
            &lt;configuration&gt;
&lt;source&gt;1.8&lt;/source&gt;
&lt;target&gt;1.8&lt;/target&gt;
            &lt;/configuration&gt;
        &lt;/plugin&gt;
        &lt;plugin&gt;
&lt;groupId&gt;org.apache.maven.plugins&lt;/groupId&gt;
;
            &lt;artifactId&gt;maven-war-
plugin&lt;/artifactId&gt;
&lt;version&gt;2.6&lt;/version&gt;
            &lt;configuration&gt;
&lt;failOnMissingWebXml&gt>false&lt;/failOnMissingWebXml&
mp;gt;
            &lt;/configuration&gt;

```

```
        &lt;/plugin&gt;  
    &lt;/plugins&gt;  
  
    &lt;/build&gt;  
&lt;/project&gt;
```

## Spring Web Initializer

El siguiente paso es crear una clase que implemente el interface de `WebApplicationInitializer`



spring web initalizer interface

```
import javax.servlet.ServletContext;  
import javax.servlet.ServletException;  
import javax.servlet.ServletRegistration;  
  
import org.springframework.web.WebApplicationInitializer;  
import  
org.springframework.web.context.support.AnnotationConfigWebApplication
```

```
Context;
import org.springframework.web.servlet.DispatcherServlet;

import com.arquitecturajava.config.ConfiguracionSpring;

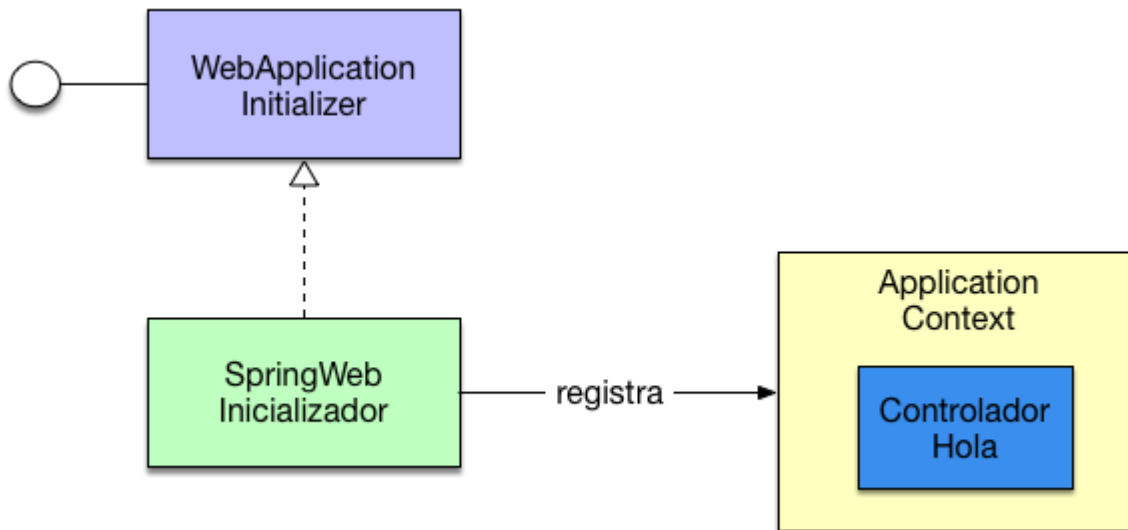
public class SpringWebInicializador implements
WebApplicationInitializer {

    public void onStartUp(ServletContext contenedor) throws
ServletException {

        AnnotationConfigWebApplicationContext contexto = new
AnnotationConfigWebApplicationContext();
        contexto.register(ConfiguracionSpring.class);
        contexto.setServletContext(contenedor);

        ServletRegistration.Dynamic servlet =
contenedor.addServlet("dispatcher", new DispatcherServlet(contexto));
        servlet.setLoadOnStartup(1);
        servlet.addMapping("/");
    }
}
```

El interface solo implementa un método onStartUp que se encarga de inicializar la aplicacion web. Para ello registra un contexto basado puramente en anotaciones.



spring web initializer y contexto

Este contexto registra un Controlador y un ViewResolver veamos el código del contexto creado con anotaciones:

```
package com.arquitecturajava.config;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.config.annotation.EnableWebMvc;
import
org.springframework.web.servlet.config.annotation.WebMvcConfigurerAdap
ter;
import
org.springframework.web.servlet.view.InternalResourceViewResolver;

@Configuration
@ComponentScan("com.arquitecturajava.*")
@EnableWebMvc
public class ConfiguracionSpring extends WebMvcConfigurerAdapter {
```

```
    @Bean
    public InternalResourceViewResolver
getInternalResourceViewResolver() {
    System.out.println("llega");
    InternalResourceViewResolver resolver = new
InternalResourceViewResolver();
    resolver.setPrefix("/WEB-INF/jsp/");
    resolver.setSuffix(".jsp");
    return resolver;
}
}
```

## Spring con @ComponentScan

En este caso estamos registrando un bean con un ViewResolver para JSP y además a través de la anotación de @ComponentScan estamos registrando clases adicionales que estén anotadas como beans de Spring dentro de com.arquitecturajava. En este ejemplo tan sencillo solo afecta al controlador.

```
package com.arquitecturajava.controllers;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class HolaController {
```

```
@RequestMapping("/hola")
public String hola() {

    return "hola";
}
}
```

Una vez configurados los beans nos queda por ver el código del fichero hola.jsp

```
&lt;html&gt;
&lt;body&gt;
hola
&lt;/body&gt;
&lt;/html&gt;
```

Acabamos de configurar una aplicación con Spring Web Initializer . Al no hacer uso de XML reduciremos de forma drástica los problemas de configuración.

Otros artículos relacionados:

**CURSO SPRING REST  
GRATIS  
APUNTATE!!**

1. [Spring Boot @Lazy Components](#)
2. [Spring @Qualifier utilizando @Autowired](#)



Spring Web Initializer ,eliminando el web.xml

3. Spring MVC Flash Attributes
4. Spring MVC