**SWS-101-JConfig:**

**TEORIA:**

**CONFIGURAZIONE CONTESTO SWS: (spring-ws-servlet.xml annotations)**

**@EnableWs: (Contesto di default)**Aggiunge di default un’ estensione della classe **WsConfigurationSupport**.  
  
Da javadocs:

This class registers the following **EndpointMappings**:

1. PayloadRootAnnotationMethodEndpointMapping ordered at 0 for mapping requests to @PayloadRoot annotated controller methods.
2. SoapActionAnnotationMethodEndpointMapping ordered at 1 for mapping requests to @SoapAction annotated controller methods.
3. AnnotationActionEndpointMapping ordered at 2 for mapping requests to @Action annotated controller methods.

Registers one **EndpointAdapter**:

1. DefaultMethodEndpointAdapter for processing requests with annotated endpoint methods.

Registers the following **EndpointExceptionResolvers**:

1. SoapFaultAnnotationExceptionResolver for handling exceptions annotated with @SoapFault.
2. SimpleSoapExceptionResolver for creating default exceptions.

**CARICAMENTO CONTESTO SWS IN CONTESTO WEB APP: (web.xml annotation)**

Classe che estende **AbstractAnnotationConfigMessageDispatcherServletInitializer**  
  
Seguendo la catena di ereditarieta si puo' verificare che estende un contesto in cui la MessageDispatcherServlet e' stata gia creata e verra' configurata con le estensioni che gli passeranno:

1) i suoi mappings (getServletMappings()==null)? AbstractMessageDispatcherServletInitializer.DEFAULT : getServletMappings()

2) config classes:

SWS Context -> getServletConfigClasses():

necessario alla MessageDispatcher

App Context -> getRootConfigClasses()

Business classes

**PRATICA:**

1. SWS ctx: Creare classe @Configuration con @EnableWs e configurare qui i componenti di SWS
2. WAC: Creare classe che extends AbstractAnnotationConfigMessageDispatcherServletInitializer:
   1. Definire qui classi AppCtx, SWSCtx, e mapping per la MessageDispatcher