**SMVC-300-Controllers**

**AFU:  
1) Controllers come form processor:**index-> Controller -> Form -> Form Processing> PageOut  
  
**2) Controller come Rest Service**Test con RestClient Mozilla

**Code Review: (debug)**

1. Giro con ModelAndView
2. Giro con MessageConverters

**TC:**

1. Abilitare configurazione @Configuration | Xml
   1. Disabilitare xml commentando servlet e servlet-mapping
   2. Disabilitare @Config commentando registerDispatcherServlet

**NOTE:**

**REST:** Impostare Accept header sul client

Form supporta solo POST? (Provare GET)

**MessageConverters:**

1. **Logica in WebMvcConfigurationSupport. getMessageConverters():**

Se non ci sono overrides (o xml) aggiunge una lista di MessageConverters di default.  
E’ possibile customizzare questa lista con:  
1.1) override configureMessageConverters(this.messageConverters); (override list)

1.2) override extendMessageConverters(this.messageConverters); (extend default)

1. **Runtime esecuzione MessageConverter (read o write o entrambi) in base a header http accept e contentType sul client, consumes, produce sul Controller.**

**Code Review:**

WebMvcConfigurationSupport. getMessageConverters()  
WebMvcConfigurationSupport.requestMappingHandlerAdapter (Code Review)

org.springframework.http.converter.json.MappingJackson2HttpMessageConverter extends

@RequestBody:  
AbstractJackson2HttpMessageConverter.readInternal  
AbstractJackson2HttpMessageConverter.getJavaType  
AbstractJackson2HttpMessageConverter.readJavaType

@ResponseBody:  
AbstractJackson2HttpMessageConverter.writeInternal

**Reflections?**In fase di boot Spring carica nel contesto degli oggetti Method e li lcollega agli handler.  
Per cui in base all uri otterra’ un oggetto Method su cui chiamera’ invoke.  
Piu lento rispetto alla chiamata diretta, ma molto piu veloce rispetto al lavoro pesante di scansione delle classi (Reflection) e dei metodi annotati fatta inizialmente.

**Performances?** Piu lento rispetto a codificare tutto via Servlet api e chiamare i metodi direttamente.  
Vantaggi, uso di framework con le sue astrazioni coi vantaggi legati a velocita sviluppo, manutenzione.