**Object domain:**

**Output Json:**

1. Method returns @ResponseBody
2. Jackson in classpath
3. Produces application/json
4. Client (Accept: application/json)

**Output Xml**

1. Classe bean annotata con @XmlRootElement
2. Xml binding lib in classpath (spring-oxm)
3. Produces application/xml
4. Client (Accept: application/xml)

**Output binario:**

1. ViewResolver per formato
2. MessageConverter e formati output binari
3. ContentNegotiator view resolver (gestione piu formati)

**BINARIES:**

1. UC: Download risorse statiche: (Blob da DB, files etc)  
   DownloadController (img, pdf):  
   <http://www.baeldung.com/spring-mvc-image-media-data> (Employee as bin)
2. UC: Download dati dinamici in formati bin (Export Excel, Export Csv):
   1. Return ModelAndView
   2. XmlViewResolver (es. Csv, Excel):  
      Mappa nome logico View con ViewResolver in file
   3. ContentNegotiationResolver:
3. UC: Uploads binari: (Multipart http da UI)
   1. Obiettivo: UploadController (img) save Employee bin -> SavePath
   2. Steps:
      1. In dispatcher.xml aggiungere un MultipartResolver
      2. Nel metodo Controller usare interface MultipartFile
4. Binaries with view resolvers:
   1. Options:
      1. By ModelAndView name (Xml
      2. By Accept Header (ContentNegotiationResolver)
   2. Steps:
      1. Add ViewResolvers in dispatcher.xml
      2. Create

**ContentNegotiationViewResolver:**Rimuovere ultimo metodo ExportController.